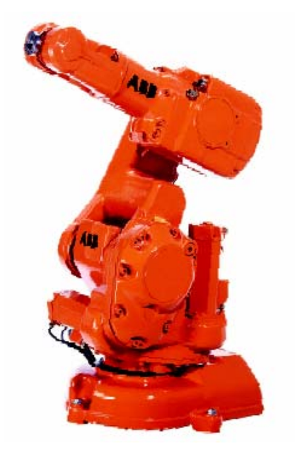
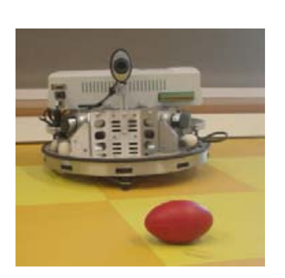
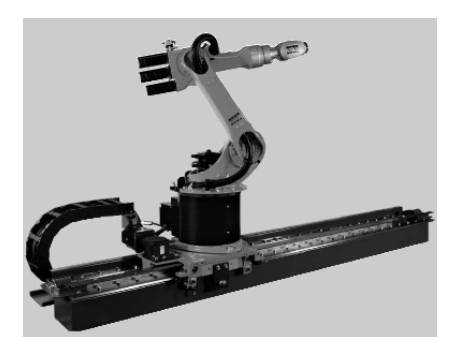
Robotique Industrielle











R. Merzouki
rochdi.merzouki@polytech-lille.fr
2010-2011

Table des Matières

I MODELISATION DES SYSTEMES ARTICULES RIGIDES 3

I.1 Introduction 4

I.2 Modélisation géométrique 4

I.2.1 Coordonnées homogènes d'un vecteur. 5

I.2.2 Paramètres de Denavit et Hartenberg 6

I.2.3 Modèle géométrique du robot PUMA 560 9

I.3 Modélisation dynamique 16

I.3.1 Approche d'Euler Lagrange 17

I.3.2 Modèle dynamique du robot PUMA 560. 25

TD : Etude de cas : robot porte-outil 28

I.4 Conclusion 36

ANNEXE I 37

ANNEXE II 41

I.5 Références 42

TP N° 1 MANIPULATION ET COMMANDE DU ROBOT A 4DDL IBM 7576 43

TP N° 2 MANIPULATION ET COMMANDE DU ROBOT A 6DDL ABB 48

TP N° 3 MANIPULATION ET COMMANDE DU ROBOT A 7DDL, KUKA 59

Chapitre I

MODELISATION DES SYSTEMES ARTICULES RIGIDES

I.1 Introduction

Le système articulé rigide est caractérisé par une structure arborescente articulé simple ou multiple dont les liaisons sont mobiles les unes par rapport aux autres. Cet ensemble a pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tâche [1].

Généralement, un robot manipulateur est considéré comme un système articulé rigide. L'appellation *robot* n'est pas nouvelle, elle a pour origine le mot *robota* extrait de la littérature grecque et qui veut dire *travail*. Nous avons trouvé dans la littérature différentes définitions de ce système dynamique tel que :

- Celle donnée par JIRA (Japon Industrial Robot Association) : *Un robot est un système versatile doté d'une mémoire et pouvant effectuer des mouvements comme ceux d'un opérateur humain* ;
- Celle donnée par RIA (Robot Institute of America) : *Un robot est un manipulateur à fonction multiple programmé pour réaliser automatiquement des taches variées éventuellement répétitives*.

La synthèse de la commande du robot nécessite la connaissance des relations entre ses grandeurs d'entrées et de sorties. L'ensemble de ces équations constitue le modèle mathématique du robot. Si les équations sont extraites de la physique, le modèle est appelé *modèle de connaissance*, et si ces équations découlent des observations disponibles sur le système, le modèle s'appelle *modèle de représentation*.

I.2 Modélisation géométrique

Tout manipulateur peut être considéré comme une chaîne de liaisons connectées par des articulations charnières ou glissières. Chaque liaison localise les informations à son propre repère. A l'aide des matrices de passages d'ordre quatre on peut arriver à des informations globales sur le repère de la base appelé repère d'inertie.

I.2.1 Coordonnées homogènes d'un vecteur.

Chaque liaison d'un manipulateur fait des rotations ou des translations par rapport au référentiel d'inertie fixe (par exemple un repère fixé à la base du robot). Le calcul des coordonnées des liaisons du manipulateur exprimées dans le référentiel d'inertie de la base est relativement difficile. Cette difficulté augmente suivant l'ordre de la liaison (numéro de la liaison) jusqu'à l'élément terminal. Pour ne pas alourdir les calculs et ramener toutes les informations géométriques au repère d'inertie de la base, il est judicieux de les localiser à leurs articulations correspondantes, et situer chaque liaison à son propre référentiel. Le passage d'un référentiel à un autre est garanti par des transformations. Lorsqu'on a uniquement des rotations on se satisfait à une matrice de transformation R de troisième ordre, et lorsqu'il existe une translation autour d'un point on est obligé de passer vers une matrice de quatrième ordre pour permettre au référentiel de se déplacer à un autre référentiel en translation. Dans ce cas le vecteur de position p sera augmenté par une quatrième composante pour avoir un vecteur de position p' exprimé par ses coordonnées homogènes :

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad \text{(I.1)}$$

Le vecteur homogène correspondant est :

$$p' = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \qquad \text{(I.2)}$$

La matrice augmentée de transformation a la forme suivante :

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & d_x \\ R_{21} & R_{22} & R_{23} & d_y \\ R_{31} & R_{32} & R_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(I.3)}$$

avec R matrice de rotation et d vecteur de translation qui comporte les coordonnées du repère destination dans le repère source.

Si T est la matrice de transformation du référentiel (x_1, y_1, z_1) vers le référentiel (x_2, y_2, z_2), alors :

$$p'_{x_2y_2z_2} = T.p'_{x_1y_1z_1} \qquad \text{(I.4)}$$

I.2.2 Paramètres de Denavit et Hartenberg

La convention de Denavit et Hartenberg (DH 1955) est une méthode systématique. Elle permet le passage entre articulations adjacentes d'un système robotique. Elle concerne **les chaînes cinématiques ouvertes où l'articulation possède uniquement un degré de liberté, et les surfaces adjacentes restent en contact**. Pour cet aspect l'utilisation des charnières ou des glissières est indispensable. Le choix adéquat des repères dans les liaisons facilite le calcul des matrices homogènes de DH et permet d'arriver à exprimer rapidement des informations de l'élément terminal vers la base ou l'inverse.

Les étapes à suivre pour cette technique sont les suivantes :

1. Numérotation des segments constitutifs du bras manipulateur de la base vers l'élément terminal. On associe le référentiel *zéro* à la base de celui-ci, et l'ordre n à l'élément terminal (effecteur) ;
2. Définition des axes principaux de chaque segment :
   - Si z_i et z_{i-1} ne se coupent pas et on choisit x_i de manière à être la parallèle avec l'axe perpendiculaire à z_i et z_{i-1} .

- Si z_i et z_{i-1} sont colinéaires on choisit x_i dans le plan perpendiculaire à z_{i-1}.

3. Fixer les quatre paramètres géométriques: d_i, q_i, a_i et θ_i (voir la figure I.1) pour chaque articulation tel que :

- d_i coordonnée de l'origine o_i sur l'axe z_{i-1}. Pour une glissière d_i est une variable et pour une charnière d_i est une constante.
- θ_i est l'angle que l'on obtient par vissage de x_{i-1} vers x_i autour de l'axe z_{i-1}. Pour une glissière q_i est une constante et pour une charnière q_i est une variable.
- a_i est la distance entre les axes z_i et z_{i-1} mesurée sur l'axe x_i négatif à partir de son origine jusqu'à l'intersection avec l'axe z_{i-1}.
- α_i est l'angle entre z_i et z_{i-1} obtenu en vissant z_{i-1} vers z_i autour de x_i.

On forme enfin la matrice homogène DH de déplacement qui lie la rotation et la translation. La partie supérieure gauche définit la matrice de rotation R_{i-1}^i, et le vecteur droit pour la translation d_{i-1}^i.

$$T_{i-1}^i = \begin{bmatrix} \multicolumn{3}{c}{R_{i-1}^i} & d_{i-1}^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.5)$$

avec

$$R_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \quad (I.6)$$

et

$$d_{i-1}^i = \begin{bmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ d_i \end{bmatrix} \quad (I.7)$$

La figure I.1 représente les paramètres de Denavit et Hartenberg pour les deux repères successifs $(x_{i-1}, y_{i-1}, z_{i-1})$ et (x_i, y_i, z_i).

Enfin, la matrice de transformation homogène de Denavit et Hartenberg est la suivante :

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.8)$$

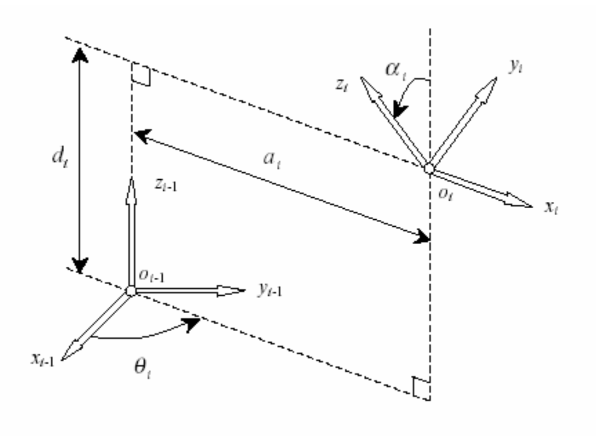


Figure I.1 : Systèmes de coordonnées et les paramètres de Denavit et Hartenberg

I.2.3 Modèle géométrique du robot PUMA 560

Parmi les différentes structures de robots manipulateurs (Robot à coordonnées : cartésienne, cylindrique, sphérique, etc.), nous avons opté pour l'architecture d'un manipulateur la plus utilisée en robotique qui est celle d'un robot articulé PUMA 560 représenté par la figure I.2. Ce bras de robot manipulateur réalise six degrés de liberté caractérisés par des mouvements de rotations



Figure I.2 : Bras de robot PUMA 560

Les trois premières articulations de ce manipulateur (Waist, Shoulder, Elbow) caractérisent pour la première une rotation autour d'un axe verticale, la second et la troisième suivant deux axes horizontaux dont les mouvements sont identifiés par les variables q_1, q_2 et q_3. Les trois dernières articulations (Wrist rotation, Wrist bend, Flange), qui constituent le poignet du bras de robot, sont caractérisées par les variables de rotations q_4, q_5 et q_6. On admet les hypothèses simplificatrices suivantes :

- Les liaisons du manipulateur sont rigides :
- Les jeux dans les articulations sont négligeables.

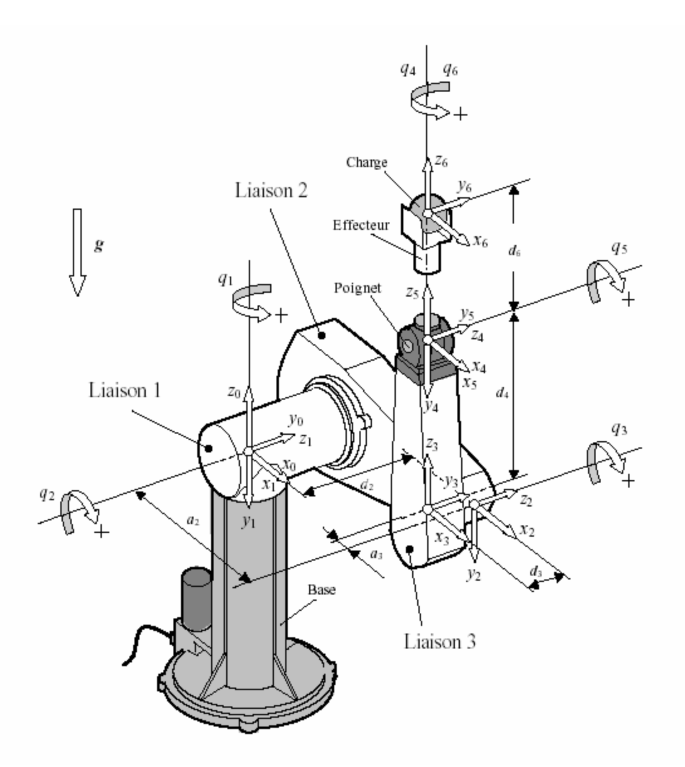


Figure I.3 : Système de coordonnées des liaisons et les paramètres des articulations pour La représentation du bras de robot PUMA 560 suivant la convention de Denavit et Hartenberg est présentée par la Figure I.3 avec une configuration d'origine.

La définition des repères associés aux segments suivant la convention de Denavit et Hatrenberg est la suivante :

Liaison 1 : Repère (x_0, y_0, z_0).

L'origine est prise dans la liaison 1 à l'intersection de l'axe de la base avec l'axe de la liaison 1.

z_0 axe de rotation, $+z_0$ vers le haut.

$+y_0$ coïncide avec l'axe de la liaison 1 et l'axe $+z_1$.

$+y_1$ est en parallèle avec la liaison 2.

Liaison 2 : Repère (x_1, y_1, z_1).

L'origine coïncide avec l'origine du repère (x_0, y_0, z_0).

z_1 axe de rotation, $+z_1$ est perpendiculaire à la liaison 2 et en parallèle à l'axe $+z_2$.

$+y_1$ vers le bas, superposé avec l'axe de la base et en parallèle avec $+y_2$.

$+x_1$ est en parallèle à la liaison 2.

Liaison 3 : Repère (x_2, y_2, z_2).

L'origine est prise dans la liaison 2 à l'intersection de l'axe de la liaison 2 avec l'axe de l'articulation 3.

z_2 axe de rotation, $+z_2$ est perpendiculaire à la liaison 2 et l'axe $+z_3$.

$+y_2$ vers le bas, opposé avec $+z_3$.

$+x_2$ est en parallèle avec la liaison 2.

Liaison 4 : Repère (x_3, y_3, z_3).

L'origine est prise dans la liaison 3.

z_3 axe de rotation, $+z_3$ vers le poignet est perpendiculaire à $+z_4$.

$+y_3$ est perpendiculaire à la liaison 2, en parallèle avec $+z_4$.

$+x_3$ est en parallèle avec la liaison 2.

Liaison 5 : Repère (x_4, y_4, z_4).

L'origine est prise dans le centre du poignet.

z_4 axe de rotation, $+z_4$ est perpendiculaire à la liaison 2 superposé avec $+z_5$.

$+y_4$ est opposé à $+z_5$.

$+x_4$ en parallèle avec la liaison 2.

Liaison 6 : Repère (x_5, y_5, z_5).

L'origine coïncide avec l'origine du repère (x_4, y_4, z_4).

z_5 axe de rotation, $+z_5$ vers l'effecteur colinéaire avec $+z_6$.

$+y_5$ coïncide avec l'axe de l'articulation 5.

$+y_5$ est perpendiculaire à l'axe de l'articulation 5.

L'effecteur : Repère (x_6, y_6, z_6).

L'origine coïncide avec les origines des repères (x_4, y_4, z_4) et (x_5, y_5, z_5)

$+z_6$ est colinéaire avec $+z_5$.

$+y_6$ est en parallèle avec $+y_5$.

$+x_6$ est en parallèle avec $+x_5$.

En respectant la position d'origine du robot et la définition des repères des liaisons présentées par la figures I.3, les paramètres du bras de robot PUMA 560 suivant la convention de Denavit et Hartenberg sont représentés dans le tableau I.1 (Armstrong *et al.*, 1986).

i Numéro de la liaison	α_i [degrés]	θ_i variable	a_i [mètres]	d_i [mètres]
1	-90	q_1	0	0
2	0	q_2	0.4318	0.2435
3	+90	q_3	-0.0203	-0.0934
4	-90	q_4	0	0.4331
5	+90	q_5	0	0
6	0	q_6	0	-

Tableau I.1 : Paramètres de Denavit et Hartenberg du bras de Robot PUMA 560.

La distance d_6 n'est pas présentée dans le tableau I.1. Cette distance varie selon l'effecteur utilisé pour l'application (l'effecteur est l'outil attaché au poignet sur la dernière articulation du robot pour la manipulation des objets : par exemple la pince représentée par la figure I.4). Dans notre application la distance entre la fin de l'effecteur et l'axe du poignet est supposée nulle $d_6 = 0$ (Armstrong *et al.*, 1986).

La dynamique des trois dernières articulations est négligeable par rapport aux trois premières. Par conséquent, nous nous sommes intéressés à étudier le mouvement des trois premières articulations du bras de robot PUMA 560 (Waist, Shoulder, Elbow), en fixant les autres à la position d'origine (i.e., poignet fixé à la position d'origine : $q_4 = 0, q_5 = 0$ et $q_6 = 0$).

Les transformations appropriées en utilisant le c pour les trois premières articulations considérées sont :

$$T_0^1 = \left[\begin{array}{ccc|c} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(I.9.a)}$$

$$T_1^2 = \left[\begin{array}{ccc|c} c_2 & -s_2 & 0 & a_2\, c_2 \\ s_2 & c_2 & 0 & a_2\, s_2 \\ 0 & 0 & 1 & d_2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(I.9.b)}$$

$$T_0^2 = \left[\begin{array}{ccc|c} c_1\, c_2 & -c_1\, s_2 & -s_1 & a_2\, c_1\, c_2 - d_2\, s_1 \\ s_1\, c_2 & -s_1\, s_2 & c_1 & a_2\, c_2\, s_1 + d_2\, c_1 \\ -s_2 & -c_2 & 0 & -a_2\, s_2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(I.9.c)}$$

$$T_2^3 = \left[\begin{array}{ccc|c} c_3 & 0 & s_3 & a_3\, c_3 \\ s_3 & 0 & -c_3 & a_3\, s_3 \\ 0 & 1 & 0 & d_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(I.9.d)}$$

$$T_0^3 = \left[\begin{array}{ccc|c} c_1\, c_{23} & -s_1 & c_1\, s_{23} & c_1\, (a_2\, c_2 + a_3\, c_{23}) - (d_2 + d_3)\, s_1 \\ s_1\, c_{23} & c_1 & s_1\, s_{23} & s_1\, (a_2\, c_2 + a_3\, c_{23}) + (d_2 + d_3)\, c_1 \\ -s_{23} & 0 & c_{23} & -(a_2\, s_2 + a_3\, s_{23}) \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad \text{(I.9.e)}$$

avec $c_i = \cos\theta_i$; $s_i = \sin\theta_i$; $c_{ij} = \cos(\theta_i + \theta_j)$; $s_{ij} = \sin(\theta_i + \theta_j)$.

A L'aide du formalisme de Denavit et Hartenberg nous pouvons calculer facilement les coordonnées de l'élément terminal (fin de l'effecteur) dans le repère (x_0, y_0, z_0) fixé à la base. La notation simplificatrice r_3^6 signifie le vecteur des coordonnées homogènes de l'élément terminal exprimées dans le repère (x_3, y_3, z_3). Avec la supposition (poignet fixé à la position d'origine :

($q_4 = 0, q_5 = 0$ et $q_6 = 0$), le vecteur r_3^6 est donné par l'équation suivante :

$$r_3^6 = \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \quad \text{(I.10)}$$

Par la transformation de Denavit et Hartenberg T_0^3, on peut calculer les coordonnées de l'élément terminal exprimées dans le repère (x_0, y_0, z_0) par la formule suivante :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = T_0^3 \, r_3^6 \qquad \text{(I.11)}$$

donc

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} c_1 c_{23} & -s_1 & c_1 s_{23} & c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 \\ s_1 c_{23} & c_1 & s_1 s_{23} & s_1 (a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 \\ -s_{23} & 0 & c_{23} & -(a_2 s_2 + a_3 s_{23}) \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \qquad \text{(I.12)}$$

Enfin, Nous définissons le vecteur position de l'élément terminal exprimé dans le repère fixé à la base (x_0, y_0, z_0) comme suit :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_1(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 + (d_4 + d_6) c_1 s_{23} \\ s_1 (a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 + (d_4 + d_6) s_1 s_{23} \\ -(a_2 s_2 + a_3 s_{23}) + (d_4 + d_6) c_{23} \end{bmatrix} \qquad \text{(I.13)}$$

A l'aide du logiciel MatLab et avec les relations du modèle géométrique du robot PUMA 560 obtenues, nous avons pu programmer un simulateur pour l'animation de ce bras de robot. Quelques fenêtres Windows de ce simulateur sont présentées dans l'annexe I.

I.3 Modélisation dynamique

Les modèles dynamiques des bras manipulateurs sont décrits par un ensemble d'équations mathématiques qui portent des informations dynamiques de ces manipulateurs et peuvent être simulées sur un ordinateur dans le but de synthétiser une commande conditionnée par des performances désirées. L'ensemble des équations dynamiques peut être déterminé par des lois mécaniques classiques Newtoniennes et Lagrangiennes. Les approches d'Euler Lagrange et Newton-Euler permettent d'aboutir aux équations du mouvement des robots.

Notion de torseur :

Un **torseur** est un objet mathématique servant en mécanique, principalement la mécanique du solide indéformable, notamment dans la modélisation des interactions entre des solides et la description de leurs mouvements.

Un torseur est constitué de deux champs vectoriels :

- un champ uniforme, dont la valeur en tout point est nommé résultante, notée $\vec{R}$;
- le champ des moments, dont la valeur en un point P est notée $\overrightarrow{M_P}$.

Ces deux champs sont reliés par la relation de Varignon : $\overrightarrow{M_P} = \overrightarrow{M_O} + \vec{R} \wedge \overrightarrow{OP}$.

Pour définir un torseur, il suffit donc de connaître sa résultante et son moment en un point. On écrit alors : $\{\tau\} = \begin{Bmatrix} \vec{R} \\ \overrightarrow{M_O} \end{Bmatrix}_O$.

Notion de tenseur :

Lorsque l'on dispose d'une base d'un espace vectoriel, tout vecteur de cet espace peut se décrire par ses coordonnées dans cette base. De même, une application linéaire entre deux espaces vectoriels, lorsque l'on a une base de chacun de ces espaces, peut se décrire par une matrice. Ainsi, dans une base $(\vec{e_1}, \vec{e_2}, \vec{e_3})$ donnée, le vecteur $\vec{u}$ sera décrit par ses composantes $(u_1,\ u_2,\ u_3)$. Si l'on change de base, les composantes (les nombres u_1, u_2 et u_3) changent, mais le vecteur $\vec{u}$ reste le même. Le tenseur représente l'ensemble des représentations de $\vec{u}$ dans toutes les bases. Un vecteur est un tenseur dit 'd'ordre 1'.

Une application linéaire f d'un espace E vers un espace F est décrite par une matrice M dont les coefficients dépendent de la base de E et de celle de F. Le tenseur représente l'ensemble des représentations de f dans toutes les bases. Une matrice est un tenseur dit 'd'ordre 2'.

Un scalaire est un simple nombre, qui ne dépend d'aucune base. On dit que le scalaire est un tenseur 'd'ordre 0'.

I.3.1 Approche d'Euler Lagrange

Le formalisme d'Euler-Lagrange et la transformation homogène de Denavit et Hartenberg amènent à un algorithme consistant à présenter les équations dynamiques du mouvement.
L'approche d'Euler-Lagrange sert à modéliser et présenter la dynamique des robots à travers les équations du mouvement. Elle s'adapte lors des calculs manuels et pour des calculs assistés par ordinateur. L'équation d'Euler-Lagrange est (Fu *et al.*, 1987 ; Paul, 1981) :

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_i}\right] - \frac{\partial L}{\partial q_i} + \frac{\partial E_D}{\partial \dot{q}_i} = T_i \qquad i = 1, \ldots, n \qquad (I.14)$$

avec E_D est l'énergie de dissipation en cas de présence de frottement visqueux, T_i est la force ou le couple généralisé à la $i^{ème}$ articulation, n est le nombre de degré de liberté, q_i est la coordonnée généralisée de l'articulation i, & $\dot{q}_i$ est la dérivée de la coordonnée généralisée et L est le Lagrangien exprimé par l'équation suivante :

$$L = E_C - E_P \qquad (I.15)$$

où E_C est l'énergie cinétique totale de toutes les liaisons, E_P est l'énergie potentielle totale de toutes les liaisons.
L'énergie cinétique est calculée par l'expression de la vitesse.

$$V_0^i = \frac{d\, r_0^i}{dt} \qquad (I.16)$$

$$r_0^i = T_0^i\, r_i^i \qquad (I.17)$$

où r_i^i est la coordonnée homogène du point i exprimée dans le repère R_i. Comme la liaison n'est pas flexible, on a $\frac{dr_i^i}{dt} = 0$ donc :

$$V_0^i = \sum_{j=1}^{i} \frac{dT_0^i}{dq_j}\frac{dq_j}{dt}\ r_i^i \qquad (I.18)$$

avec T_{i-1}^i est la matrice de transformation homogène. Sous une écriture matricielle, on peut écrire :

$$V_0^i = \sum_{j=1}^{i}\left[U_{ij}\dot{q}_j\right] r_i^i \qquad (I.19)$$

et

$$U_{ij} = \begin{cases} T_0^{j-1} Q_j T_{j-1}^i & j \le i \\ 0 & j \ge i \end{cases} \qquad (I.20)$$

Pour une liaison rotoïde, on a :

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (I.21)$$

Et pour une liaison prismatique :

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (I.22)$$

L'énergie cinétique de l'élément i dans la liaison i est

$$dE_{Ci} = \frac{1}{2} trace\left(V_i V_i^{\mathrm{T}}\right) dm \qquad (I.23)$$

En développant l'expression précédente, on obtient :

$$dE_{Ci} = \frac{1}{2} trace\left(\sum_{j=1}^{i} \sum_{k=1}^{i} U_{ij} \left(r_i^i r_i^{i\mathrm{T}} dm \right) U_{ik}^{\mathrm{T}} \dot{q}_j \dot{q}_k \right) \qquad (I.24)$$

L'énergie cinétique de la liaison i est :

$$E_{Ci} = \frac{1}{2} trace\left(\sum_{j=1}^{i} \sum_{k=1}^{i} U_{ij} J_i U_{ik}^{\mathrm{T}} \dot{q}_j \dot{q}_k \right) \qquad (I.25)$$

avec

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i^2 dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \qquad (I.26)$$

L'énergie cinétique des actionneurs est définie par :

$$E_{Ca} = \frac{1}{2} \sum_{i=1}^{n} I_i \dot{q}_i^2 \qquad (I.27)$$

où I_i caractérise un moment d'inertie dans le cas d'une rotation et une masse dans le cas d'une translation de l'actionneur i. L'énergie cinétique totale sera :

$$E_C = \sum_{i=1}^{n} E_{Ci} + E_{Ca} \qquad (I.28)$$

L'énergie potentielle est décrite par (Fu *et al.*, 1987)

$$E_P = \sum_{i=1}^{n} -m_i g^{\mathrm{T}} r_0^i = -\sum_{i=1}^{n} m_i g^{\mathrm{T}} T_0^i r_i^i \qquad (I.29)$$

avec

$$g^{\mathrm{T}} = \begin{bmatrix} g_x & g_y & g_z & 1 \end{bmatrix}$$

exprimée dans le repère R_0. Dans le cas où l'axe z_0 est orienté vers le haut, nous obtiendrons :

$$g^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & -|g| & 1 \end{bmatrix} \qquad (I.30)$$

où g est la gravité mesuré à la base du robot.

Enfin, l'énergie de dissipation est donnée par :

$$E_D = \frac{1}{2} \sum_{i=1}^{n} f_{Vi} \dot{q}_i^2 \qquad (I.31)$$

où f_{vi} le coefficient de frottement visqueux (i=1,n). L'expression finale du Lagrangien sera :

$$L = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} trace\left(U_{ij} J_i U_{ik}^{\mathrm{T}} \right) \dot{q}_j \dot{q}_k + \sum_{i}^{n} m_i g^{\mathrm{T}} T_0^i r_i^i \qquad (I.32)$$

Appliquant la formule d'Euler-Lagrange à la fonction Lagrangienne donnée ci-dessus, on trouve la force ou couple généralisé

$$\begin{aligned} T_i = & \sum_{j=1}^{n} \sum_{k=1}^{j} trace\left(U_{jk} J_j U_{ji}^{\mathrm{T}} \right) \ddot{q}_k \\ & + \sum_{j=1}^{n} \sum_{k=1}^{j} \sum_{l=1}^{j} trace\left(U_{jkl} J_j U_{ji}^{\mathrm{T}} \right) \dot{q}_k \dot{q}_l \\ & - \sum_{j=1}^{n} m_i g^{\mathrm{T}} U_{ji} r_j^j + f_{Vi} \dot{q}_i \end{aligned} \qquad (I.33)$$

avec

$$U_{ijk} = \begin{cases} T_0^{k-1} Q_k T_{k-1}^{j-1} Q_j T_{j-1}^i & k \le j \le i \\ T_0^{j-1} Q_j T_{j-1}^{k-1} Q_k T_{k-1}^i & j \le k \le i \\ 0 & j < i < k \end{cases} \quad (I.34)$$

Avec une autre présentation, l'équation (I.33) devienne :

$$T_i = \sum_{j=1}^{n} M_{ij}(q)\,\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} N_{ijk}(q)\,\dot{q}_j\,\dot{q}_k + G_i(q) + H_i(\dot{q}_i), \qquad i = 1,\ldots,n \quad (I.35)$$

avec

$$M_{ij}(q) = \sum_{k=\max(i,j)}^{n} trace\left(U_{kj}\; J_k\; U_{ki}^{\mathrm{T}}\right) \qquad \begin{matrix} i = 1,\ldots,n \\ j = 1,\ldots,n \end{matrix} \quad (I.36)$$

$$G_i(q) = -\sum_{j=i}^{n} m_j\, g^{\mathrm{T}}\, U_{ji}\, r_j^j \quad (I.37)$$

$$N_{ijk}(q) = \sum_{l=\max(i,j,k)}^{n} trace\left(U_{ljk}\; J_l\; U_{li}^{\mathrm{T}}\right) \qquad \begin{matrix} i = 1,\ldots,n \\ j = 1,\ldots,n \\ k = 1,\ldots,n \end{matrix} \quad (I.38)$$

$$H_i(\dot{q}_i) = f_{Vi}\;\dot{q}_i \quad (I.39)$$

L'écriture matricielle des équations précédentes donne le modèle dynamique global du robot (Seraji, 1988). Ce modèle est le suivant :

$$T(t) = M(q)\,\ddot{q} + N(q,\dot{q}) + G(q) + H(\dot{q}). \quad (I.40)$$

avec $q \in R^n, \dot{q} \in R^n, \ddot{q} \in R^n$ représentent respectivement les positions, les vitesses et les accélérations articulaires; $M(q) \in R^{n\times n}$ est la matrice symétrique définie positive des accélérations inertielles dont l'élément M_{ij} de cette matrice est l'inertie de la $i^{ème}$ articulation sur la $j^{ème}$ articulation et vice versa; $N(q,\dot{q}) \in R^n$ est le vecteur de forces et/ou couples dus aux accélérations de Coriolis et centrifuge;

$G(q) \in R^n$ est le vecteur de forces et/ou couples dus aux forces de gravitation; $H(\dot{q}) \in R^n$ représente les effets des frottements visqueux; $T(t) \in R^n$ est le vecteur de forces et/ou couples moteurs.

Une autre forme de représentation matricielle du modèle dynamique (I.35) est la suivante (Armstrong, 1986) :

$$T(t) = M(q)\,\ddot{q} + D(q)\left[\dot{q}\;\dot{q}\right] + C(q)\left[\dot{q}^2\right] + G(q) + H(\dot{q}). \quad (I.41)$$

avec $q, \dot{q}, \ddot{q} \in R^n$ représentent respectivement les positions, les vitesses et les accélérations articulaires;

$M(q) \in R^{n\times n}$ est la matrice symétrique définie positive des accélérations inertielles dont l'élément M_{ij} de cette matrice est l'inertie de la $i^{ème}$ articulation sur la $i^{ème}$ articulation et vice versa ; $D(q) \in R^{n\times\frac{n(n-1)}{2}}$ est la matrice des couples de Coriolis ; $C(q) \in R^{n\times n}$ est la matrice des couples de centrifuges ; $G(q) \in R^n$ est le vecteur des accélérations gravitationnelles ; le vecteur $H(\dot{q}) \in R^n$ représente les effets des frottements visqueux ; $T(t) \in R^n$ est le vecteur de forces ou couples généralisés ; $[\dot{q},\dot{q}] \in R^{\frac{n(n-1)}{2}}$ est le vecteur de produit des vitesses généralisées et $\left[\dot{q}^2\right] \in R^n$ est le vecteur de carré des vitesses généralisées, tel que :

$$\left[\dot{q}\;\dot{q}\right] = \left[\dot{q}_1\dot{q}_2 \quad \ldots \quad \dot{q}_1\dot{q}_n \quad \dot{q}_2\dot{q}_3 \quad \ldots \quad \dot{q}_2\dot{q}_n \quad \ldots \quad \dot{q}_{n-2}\dot{q}_n \quad \dot{q}_{n-1}\dot{q}_n\right]^{\mathrm{T}} \quad (I.42)$$

et

$$\left[\dot{q}^2\right] = \left[\dot{q}_1^2 \quad \dot{q}_2^2 \quad \ldots \quad \dot{q}_n^2\right]^{\mathrm{T}} \quad (I.43)$$

Il existe une méthode directe basée sur le principe de conservation d'énergie pour calculer rapidement la matrice des couples de Coriolis D et la matrice des couples centrifuges C. Cette méthode est la suivante (Armstrong *et al.*, 1986)

Pour la matrice de Coriolis D, le calcul est basé essentiellement sur la formule suivante :

$$D_{ij} = 2\beta^{\,i,kl} \quad (I.44)$$

avec le symbole de Christoffel défini par :

$$\beta^{i,jk} = \frac{1}{2}\left[\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i}\right] \quad (I.45)$$

Puisque la matrice d'inertie est définie positive, on aura les propriétés suivantes :

$$\frac{\partial m_{ij}}{\partial q_k} = \frac{\partial m_{ji}}{\partial q_k} \qquad \forall\, i, j, k \quad (I.46)$$

et

$$\frac{\partial m_{ij}}{\partial q_k} = 0 \qquad \text{pour} \quad i \geq k, \quad j \geq k \quad (I.47)$$

Enfin, les éléments de la matrice centrifuge C sont définis par l'équation suivante :

$$C_{ij} = \beta^{i,jj} \quad (I.48)$$

Le vecteur des frottements visqueux et secs $H(\dot{q})$ peut être représenté par les n

éléments

$h_i(\dot{q}_i),\ i = 1,\ldots,n$ sous la forme suivante :

$$h_i(\dot{q}_i) = f_{Vi}\,\dot{q}_i + f_{Si}\,sgn(\dot{q}_i) \quad (I.49)$$

avec f_{Vi} et f_{Si} sont respectivement les coefficients des frottements visqueux et secs de la i ^{ème} articulation.

La perturbation due à la portée de la charge représentée par U_m caractérise l'effet de la charge m sur le modèle dynamique du robot, il est calculé par la matrice Jacobienne. Celle-ci est la dérivée du vecteur position de l'élément terminal (fin effecteur). Elle est donnée par la formule suivante :

$$J(q) = \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \cdots & \frac{\partial p_x}{\partial q_n} \\ \frac{\partial p_y}{\partial q_1} & \cdots & \frac{\partial p_y}{\partial q_n} \\ \frac{\partial p_z}{\partial q_1} & \cdots & \frac{\partial p_z}{\partial q_n} \end{bmatrix} \quad (I.50)$$

La perturbation due à la portée de la charge sera :

$$u_m = m\ J^T(q)\left[J(q)\,\ddot{q} + \dot{J}(q,\dot{q})\,\dot{q} + g\right] \quad (I.51)$$

avec g le vecteur de gravité mesuré sur base du robot.

I.3.2 Modèle dynamique du robot PUMA 560.

Dans cette section, le modèle dynamique du bras de robot PUMA 560 est présenté. Les trois dernière articulations de ce manipulateurs qui constituent le poignet (Wrist) sont fixées à la position *zéro* (i.e., les positions $q_4 = 0, q_5=0$ et $q_6 = 0$, les vitesses $\dot{q}_4 = 0, \dot{q}_5=0$ et $\dot{q}_6 = 0$ et les accélérations $\ddot{q}_4 = 0, \ddot{q}_5=0$ et $\ddot{q}_6 = 0$, $\forall t \in R^{+}{}_{+}\Re \in \forall t$). Nous avons adopté l'abréviation des fonctions trigonométriques des coordonnées articulaires du robot en écrivant c_i, s_i, c_{ij}, s_{ij} *et* c_{ijk} pour signifier $\cos q_i, \sin q_i, \cos(q_i + q_j)$, *et* $\cos(q_i + q_j + q_k)$ respectivement.

Le modèle dynamique pour les trois premières articulations du bras de robot PUMA 560 est le suivant :

$$M(q)\,\ddot{q} + B(q,\dot{q})\,\dot{q} + G(q) = u - m\,J^T(q)\left[J(q)\,\ddot{q} + \dot{J}(q,\dot{q})\,\dot{q} + g\right] \qquad \text{(I.52)}$$

avec $q = [q_1, q_2, q_3]^T$ le vecteur des coordonnées généralisées, *m* est le poids de la charge à

porter, $J(q)$ la matrice Jacobienne et $q = [q_1, q_2, 9.81]^T$.

Les expressions des éléments des matrices *M, B* et le vecteur *G* sont les suivants :

- Les éléments de la matrice d'inertie *M*(*q*) :

$$\begin{aligned}
m_{11} &\approx 2.57 + 1.38\,c_2^2 + 0.30\,s_2\,s_3 + 7.44\times10^{-1}\,c_2\,s_{23}\\
m_{12} = m_{21} &\approx 6.90\times10^{-1}\,s_2 - 1.34\times10^{-1}\,c_{23} + 2.38\times10^{-2}\,c_2\\
m_{13} = m_{31} &\approx -1.34\times10^{-1}\,c_{23} - 3.97\times10^{-3}\,s_{23}\\
m_{22} &\approx 6.79 + 7.44\times10^{-1}\,s_3\\
m_{23} = m_{32} &\approx 0.333 + 3.72\times10^{-1}\,s_3 - 1.10\times10^{-2}\,c_3\\
m_{33} &\approx 1.16
\end{aligned} \qquad \text{(I.53)}$$

- Les éléments le la matrice $B(q,\dot{q})$:

$$\begin{aligned}
b_{11} &\approx (-2.76\,s_2\,c_2 + 7.44\times10^{-1}\,c_{223} + 0.60\,s_2\,c_3 - 2.13\times10^{-2}\,(1 - 2\,s_2\,s_3))\,\dot{q}_2\\
b_{12} &\approx (6.90\times10^{-1}\,c_2 + 1.34\times10^{-1}\,s_{23} - 2.38\times10^{-2}\,s_2)\,\dot{q}_2\\
&\quad + (2.67\times10^{-1}\,s_{23} - 7.58\times10^{-3}\,c_{23})\,\dot{q}_3\\
b_{13} &\approx (7.44\times10^{-1}\,c_2\,c_{23} + 0.60\,s_2\,c_3 + 2.20\times10^{-2}\,c_2\,s_{23}\\
&\quad - 2.13\times10^{-2}\,(1 - 2\,s_2\,s_3))\,\dot{q}_1 + 0.5\,(2.67\times10^{-1}\,s_{23} - 7.58\times10^{-3}\,c_{23})\,\dot{q}_3\\
b_{21} &\approx -0.5\,(-2.76\,s_2\,c_2 + 7.44\times10^{-1}\,c_{223} + 0.60\,s_2\,c_3 - 2.13\times10^{-2}\,(1 - 2\,s_2\,s_3))\,\dot{q}_1\\
b_{22} &\approx (2.20\times10^{-2}\,s_3 + 7.44\times10^{-1}\,c_3)\,\dot{q}_3\\
b_{23} &\approx 0.5\,(2.20\times10^{-2}\,s_3 + 7.44\times10^{-1}\,c_3)\,\dot{q}_3\\
b_{31} &\approx -0.5\,(7.44\times10^{-1}\,c_2\,c_{23} + 0.60\,s_2\,c_3\\
&\quad + 2.20\times10^{-2}\,c_2\,s_{23} - 2.13\times10^{-2}\,(1 - 2\,s_2\,s_3))\,\dot{q}_1\\
b_{32} &\approx -0.5\,(2.20\times10^{-2}\,s_3 + 7.44\times10^{-1}\,c_3)\,\dot{q}_2\\
b_{33} &= 0
\end{aligned} \qquad \text{(I.54)}$$

- Les élément du vecteur de gravité $G(q)$:

$$\begin{aligned}
g_1 &= 0\\
g_2 &\approx -37.2\,c_2 - 8.4\,s_{23} + 1.02\,s_2\\
g_3 &\approx -8.4\,s_{23} + 0.25\,c_{23}
\end{aligned} \qquad \text{(I.55)}$$

La démonstration et le calcul des éléments des matrices *M, B* et le vecteur *G* du modèle dynamique de ce bras de robot manipulateur est présenté dans la référence (Armstrong *et al.*,1986).

La matrice Jacobienne *J*(*q*) est obtenue par les formules (I.13) et (I.51). Après quelques calcules simples nous trouvons les expressions des éléments de la matrice *J*(*q*) :

$$
\begin{aligned}
j_{11} &= -s_1(a_2\,c_2 + a_3\,c_{23}) - (d_2 + d_3)\,c_1 - (d_4 + d_6)\,s_1\,s_{23} \\
j_{12} &= -c_1\,(a_2\,s_2 + a_3\,s_{23}) + (d_4 + d_6)\,c_1\,c_{23} \\
j_{13} &= -a_3\,c_1\,s_{23} + (d_4 + d_6)\,c_1\,c_{23} \\
j_{21} &= c_1\,(a_2\,c_2 + a_3\,c_{23}) - (d_2 + d_3)\,s_1 + (d_4 + d_6)\,c_1\,s_{23} \\
j_{22} &= -s_1\,(a_2\,s_2 + a_3\,s_{23}) + (d_4 + d_6)\,s_1\,c_{23} \\
j_{23} &= -a_3\,s_1\,s_{23} + (d_4 + d_6)\,s_1\,c_{23} \\
j_{31} &= 0 \\
j_{32} &= -(a_2\,c_2 + a_3\,c_{23}) - (d_4 + d_6)\,s_{23} \\
j_{33} &= -a_3\,c_{23} - (d_4 + d_6)\,s_{23}
\end{aligned}
\qquad (I.56)
$$

avec a_2 = 0.4319 m, a_3 = –0.0203 m, b_2 = 0.2435 m, b_3 = –0.0934 m, b_4 = 0.4331 m et b_6 = 0 m.
Les paramètres physiques du bras de robot PUMA 560 sont représentés dans l'annexe II.

Pour plus d'informations voir (Armstrong *et al.*, 1986). Le tableau Ann.II.1 représente les masses des différentes liaisons du manipulateur ; la masse de la première liaison n'est pas présentée dans ce tableau puisque cette liaison ne fait pas une translation par rapport à la base du robot mais seulement une rotation de l'articulation 1 autour de l'axe z_0 (énergie potentielle nulle). La mesure de cette masse n'est pas nécessaire. Les différents moments d'inerties relatifs à chaque liaison en respectant les repères des articulations sont représentés par Le tableau Ann.II.2. La notation simplificatrice I_{xx}, I_{yy} et I_{zz} signifi respectivement le moment d'inertie suivant l'axe x, y et z relatif à chaque articulation, et la notation I_{moteur} le moment d'inertie du moteur. Le rapport d'engrenage et le couple maximum de chaque articulation sont présentés par le tableau Ann.II.3.

Etude de Cas d'un robot porte-outil

(Problème d'Examen de DEA Robotique 92-93 UMPC-ENSAM, cours A. Barraco)

Le robot étudié est schématisé par un ensemble de solides représenté par la Figure 1. Le corps 0 de référence (C_0) est repéré par le repère $R_0(O_0,\vec{x_0},\vec{y_0},\vec{z_0})$ de type galiléen. Le corps 1 (C_1) a un mouvement de rotation autour de l'axe $(O_0,\vec{z_0})$, sachant que le repère lié à ce corps est $R_1(O_1,\vec{x_1},\vec{y_1},\vec{z_1})$. La position relative de (C_1) par rapport à (C_0) est $q_1 = (\vec{x_0},\vec{x_1})$. Le corps (C_2) a un mouvement de rotation autour de l'axe $(O_2,\vec{z_2})$, colinéaire avec l'axe $(O_1,\vec{x_1})$ avec $\overrightarrow{O_1O_2} = -a\vec{y_1}$. Le repère lié à ce corps est $R_2(O_2,\vec{x_2},\vec{y_2},\vec{z_2})$. La position relative de (C_2) par rapport à (C_1) est $q_2 = (\vec{x_1},\vec{x_2})$. Le corps (C_3) réalise par rapport à (C_2) un mouvement de translation d'axe $(O_2,\vec{y_2})$ défini par $\overrightarrow{O_2O_3} = q_4\vec{y_2}$ et un mouvement de rotation autour de $(O_2,\vec{y_2})$ avec un angle $q_3 = (\vec{x_2},\vec{x_3})$. Le repère lié à ce corps est $R_3(O_3,\vec{x_3},\vec{y_3},\vec{z_3})$. Les caractéristiques massiques, inertielles et géométriques sont indiquées dans la Figure 1.
Le point $\boldsymbol{P_3}$ appartient à l'organe terminal de (C_3), tel que : $\overrightarrow{O_3P_3} = b\vec{z_3}$
L'objet de ce problème est de faire l'étude géométrique, cinématique et dynamique du robot.

1. ETUDE GEOMETRIQUE

1.1 Question : position du point $\boldsymbol{P_3}$

Exprimer dans le repère $R_1(O_1,\vec{x_1},\vec{y_1},\vec{z_1})$, le vecteur $\overrightarrow{O_1P_3}$.

Réponse :

1.2 Question : modèle géométrique inverse.

On souhaite que le point $\boldsymbol{P_3}$ occupe une position particulière dans l'espace, ce qui revient à imposer $\overrightarrow{O_1P_3} = \lambda\vec{u}$, avec $\vec{u}$ vecteur unitaire non nul dont on connaît les projections dans le repère $R_0(O_0,\vec{x_0},\vec{y_0},\vec{z_0})$; $\vec{u} = u_x\vec{x_0} + u_y\vec{y_0} + u_z\vec{z_0}$ avec $u_x, u_y, u_z \neq 0$ et λ scalaire non nul.

Déterminer le ou les jeux de paramètres, $\boldsymbol{q}$, qui sont solutions. On commencera par déterminer $\boldsymbol{q_1}$ puis les autres paramètres.

Réponse :

2. ETUDE CINEMATIQUE

2.1 Question : Vitesse de Rotation.

Donner les expressions des vitesses de rotation de chaque corps $\vec{\omega}(i/0)$ **avec** $i \in [1,3]$, exprimées pour $\vec{\omega}(1/0)$ **dans** $R_1\left(O_1, \overrightarrow{x_1}, \overrightarrow{y_1}, \overrightarrow{z_1}\right)$, pour $\vec{\omega}(2/0)$ **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$ et pour $\vec{\omega}(3/0)$ **dans** $R_3\left(O_3, \overrightarrow{x_3}, \overrightarrow{y_3}, \overrightarrow{z_3}\right)$ et **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Donner les expressions des dérivées galiléennes des vecteurs de la base $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Réponse :

2.2 Question : Vitesses et accélération des différents points.

Donner les expressions des vitesses galiléennes des points : O_2, O_3 **et** P_3 exprimées **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Donner l'expression de l'accélération galiléenne de O_3 exprimée **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Réponse :

2.3 Question : Cinématique inverse.

On suppose que $b + q_4 \neq 0$. Peut-on trouver des jeux de paramètres q et $\dot{q}$ tels que $\vec{V}(P_3, 3/0) = \vec{u}$, où $\vec{u}$ est un vecteur imposé dont on connaît les composantes **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$?

Réponse :

2.4 Question : Cinétique.

Donner les expressions des moments cinétiques $\vec{K}(O_1, 1/0), \vec{K}(O_2, 2/0), \vec{K}(O_3, 3/0)$, exprimé pour le premier **dans** $R_1\left(O_1, \overrightarrow{x_1}, \overrightarrow{y_1}, \overrightarrow{z_1}\right)$ et pour les deux autres **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Donner les expressions des moments dynamiques $\vec{H}(O_2, 2/0), \vec{H}(O_3, 3/0)$, exprimés **dans** $R_2\left(O_2, \overrightarrow{x_2}, \overrightarrow{y_2}, \overrightarrow{z_2}\right)$.

Calculer les énergies cinétiques de chaque corps $T_i(i/0)$ **avec** $i \in [1, 3]$.

Réponse :

3. ETUDE DYNAMIQUE

Chaque liaison est motorisée. Entre le corps (C_0) **et** (C_1) un moteur exerce sur le corps (C_1) un moment $\overrightarrow{\tau_1} = \tau_1 \overrightarrow{z_1}$, et sur le corps (C_0) un moment opposé. Entre le corps (C_1) **et** (C_2) un moteur exerce sur le corps (C_2) un moment $\overrightarrow{\tau_2} = \tau_2 \overrightarrow{z_2}$, et sur le corps (C_1) un moment opposé. Entre le corps (C_2) **et** (C_3) un moteur exerce sur le corps (C_3) un moment $\overrightarrow{\tau_3} = \tau_3 \overrightarrow{z_3}$ et un effort $\overrightarrow{t_3} = t_3 \overrightarrow{z_3}$, et sur le corps (C_2) des efforts opposés.

On considère également en P_3 un torseur $\overrightarrow{F_3}; \overrightarrow{M_3}(P_3)$, représentent les actions extérieures à l'outil.

3.1 Question : Puissance des efforts extérieurs.

Calculer la puissance P_{mot} des efforts des moteurs.

Donner l'expression de la puissance P_{ext} des efforts extérieurs en P_3, en fonction de $\overrightarrow{V}(P_3)$ et $\vec{\omega}(3/0)$, on ne développera pas cette expression.

Réponse :

3.2 Question : Equations de Newton-Euler.

Ecrire l'équation de Newton pour le corps (C_3), projetée suivant la direction $\overrightarrow{z_3}$.

Ecrire l'équation d'Euler pour le corps (C_3), projetée suivant la direction $\overrightarrow{z_3}$.

Réponse :

3.3 Question : Formalisme de Lagrange.

A partir de l'expression de l'énergie cinétique calculée à la question 2.4 et en utilisant le formalisme de Lagrange, retrouver les deux équations du mouvement de la question 3.2.

Réponse :

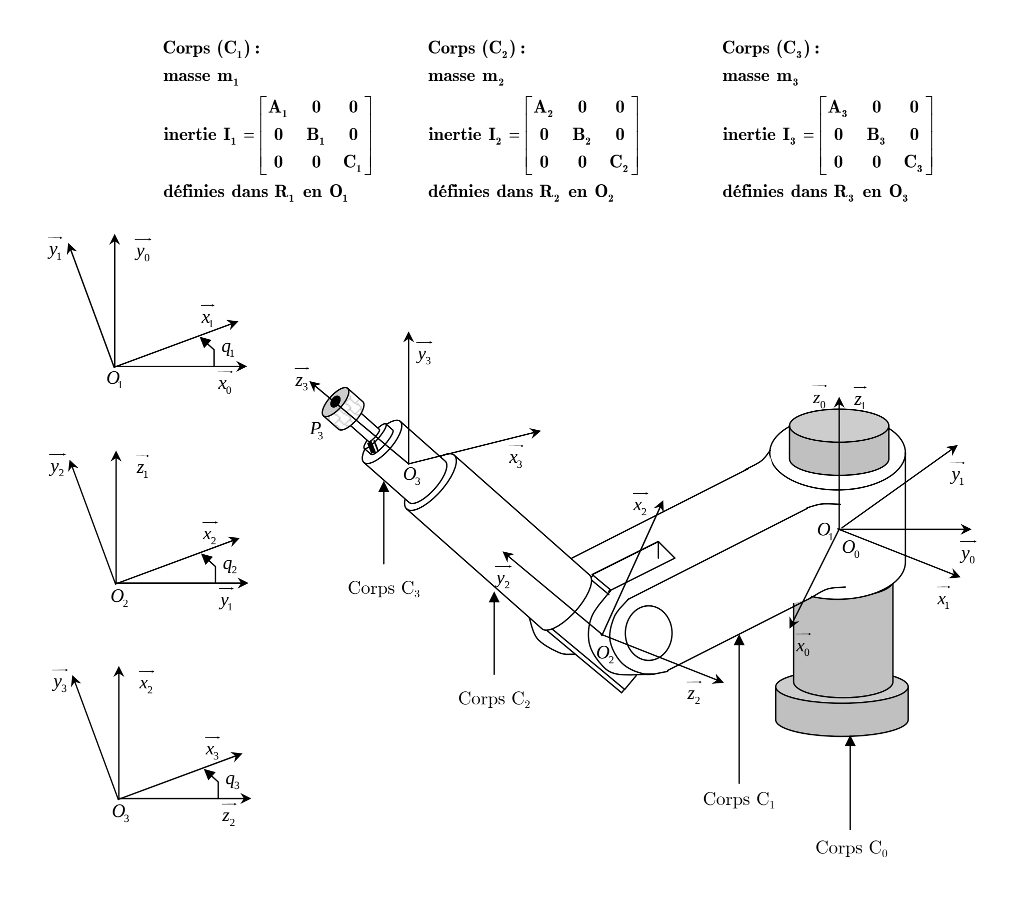


Figure 1 : Description du robot

I.4 Conclusion

Les formalismes de Lagrange Euler permettent d'établir un système d'équations différentielles reliant les coordonnées généralisées aux forces et/ou couples généralisés. Cette procédure nous a permit d'obtenir les modèles dynamiques du robot PUMA 560. Le modèle a permis de mettre en évidence la difficulté de commander ce genre de bras manipulateur, étant donné les fortes non-linéarités qui régissent le fonctionnement dynamique de ceux-ci.

Nous avons remarqué que le comportement dynamique du robot manipulateur exige une trajectoire spécifique (continue en position, en vitesse et en accélération), permettant ainsi le contrôle du robot avec des commandes lisses et physiquement acceptables.

ANNEXE I

ANIMATION DU ROBOT PUMA 560

Nous avons réalisé un programme en MatLab pour animer le bras de robot PUMA 560 avec une vision 3D. Ce programme nous a permis de visualiser les différentes configurations du robot obtenues par les simulations. La figure Ann.I.1 représente la fenêtre du menu. A l'aide de cette dernière, on peut sélectionner les conditions initiales, conditions finales, temps de mouvement et le type des trajectoires de références.

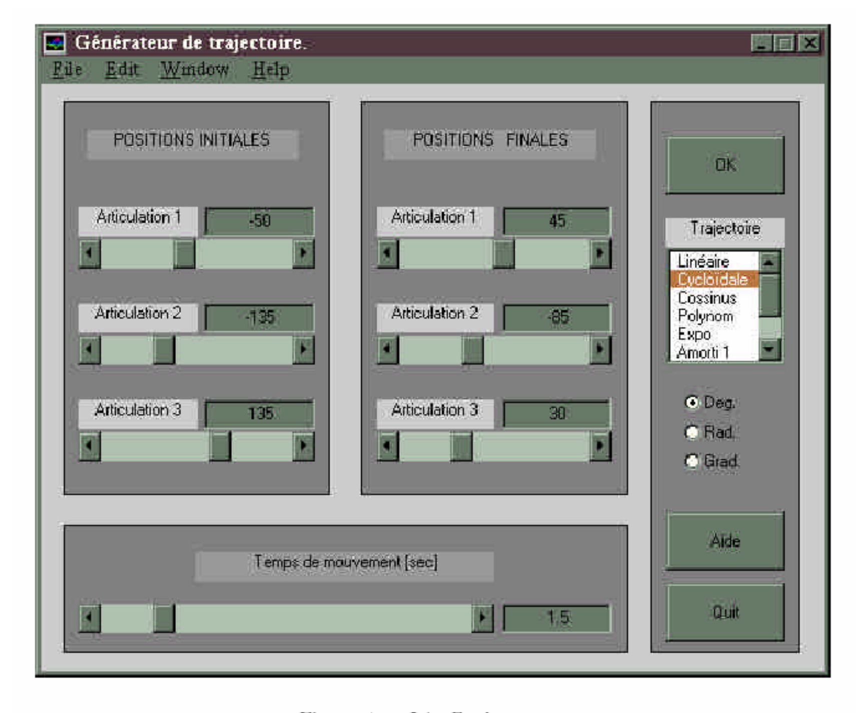


Figure Ann.I.1 : Fenêtre menu.

La figure Ann.I.2 représente une image d'une animation 3D du robot PUMA 560. On peut également faire des projections de l'animation suivant les deux plans XZ et XY (voir les figures Ann.I.3 et Ann.I.4).

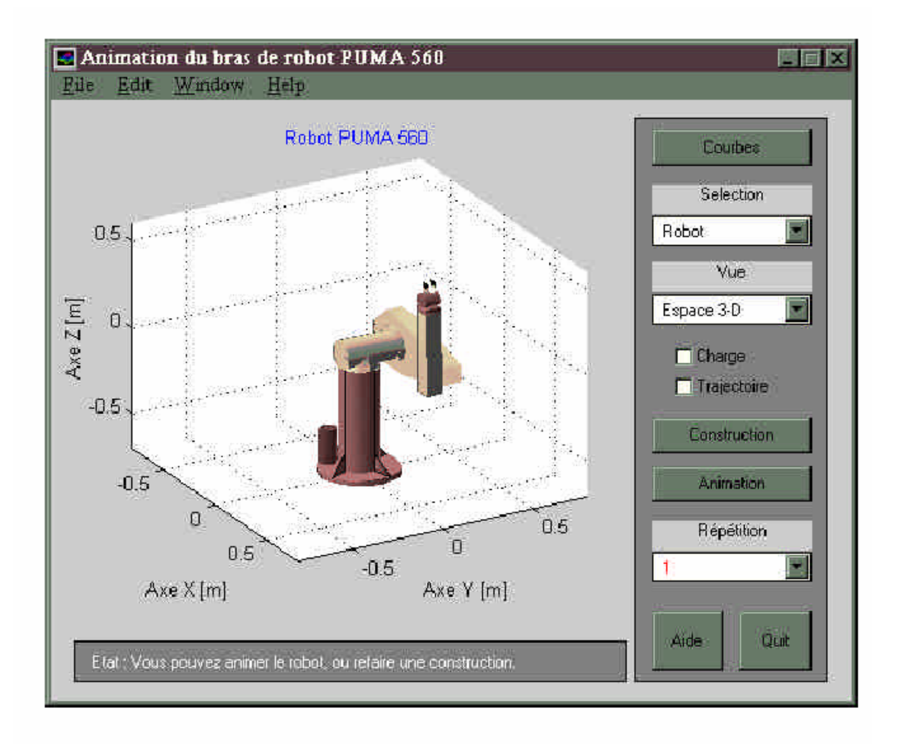


Figure Ann.I.2 : Animation du robot PUMA 560 en 3D.

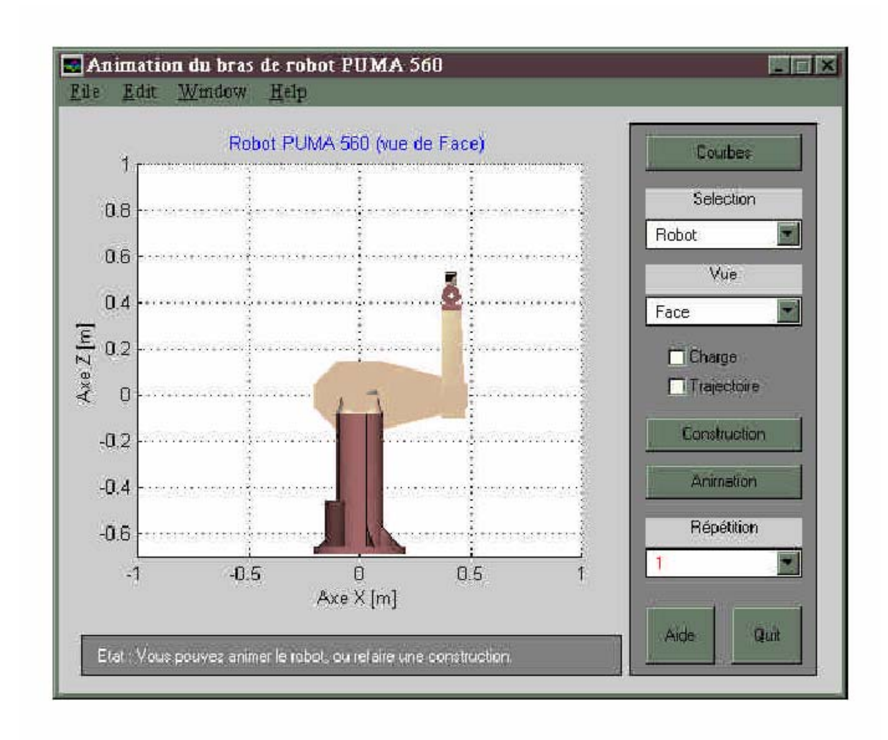


Figure Ann.I.3 : Animation du robot PUMA 560 dans le plan XZ.

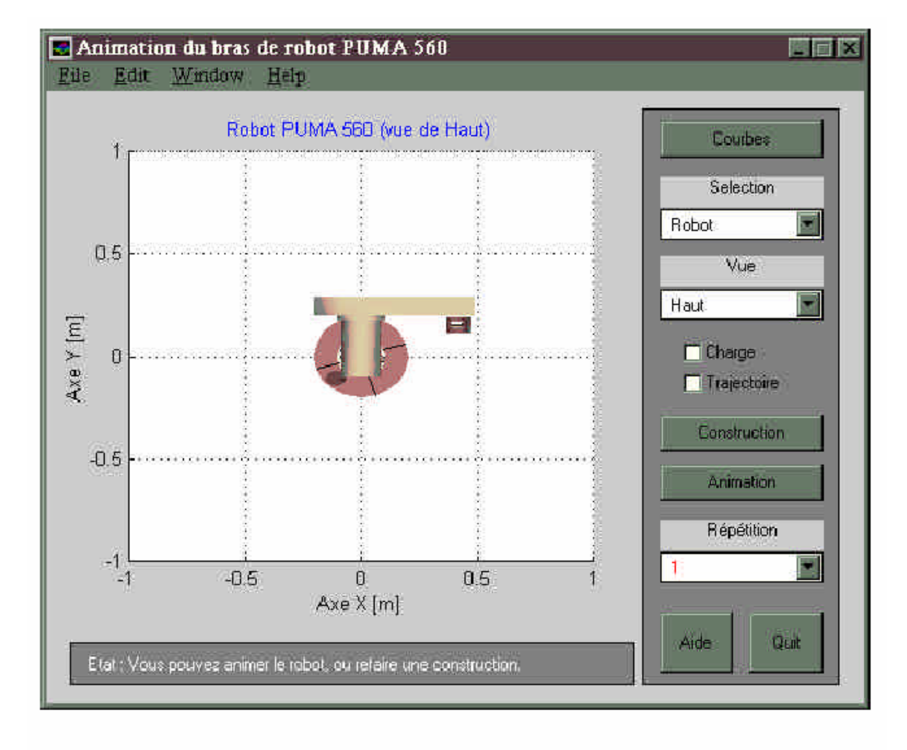


Figure Ann.I.4 : Animation du robot PUMA 560 dans le plan XY.

ANNEXE II

PARAMETRES REELS DU ROBOT PUMA 560

Parties	Poids (kg)
Liaison 1	-
Liaison 2	17.40
Liaison 3	4.80
Liaison 3 avec Poignet	6.04
Liaison 4 *	0.82
Liaison 5 *	0.34
Liaison 6 *	0.09
Poignet	1.24

* Valeurs avec tolérance (kilogrammes ± 25%)

Tableau Ann.II.1 : Poids des déférentes liaisons (kilogrammes : ±0.01 + 1%).

Parties	I_{xx}	I_{yy}	I_{zz}	I_{moteur}
Liaison 1	-	-	0.35	1.14 (± 0.27)
Liaison 2	0.130 (± 3%)	0.524 (± 5%)	0.539 (± 3%)	4.71 (± 0.54)
Liaison 3	0.066	0.0125	0.086	0.83 (± 0.09)
Liaison 3 Avec Poignet	0.192 (± 4%)	0.0154 (± 5%)	0.212 (± 4%)	-
Liaison 4 *	$1.80 \; 10^{-3}$	$1.80 \; 10^{-3}$	$1.30 \; 10^{-3}$	0.200 (± 0.016)
Liaison 5 *	$0.30 \; 10^{-3}$	$0.30 \; 10^{-3}$	$0.40 \; 10^{-3}$	0.179 (± 0.014)
Liaison 6 *	$0.15 \; 10^{-3}$	$0.15 \; 10^{-3}$	$0.04 \; 10^{-3}$	0.193 (± 0.015)

* Valeurs avec tolérence : ± 50%

Tableau Ann.II.2 : Paramètres d'inertie (kg·m²).

Numéro de l'articulation	1	2	3	4	5	6
Rapport d'engrenage	62.61	107.36	53.69	76.01	71.91	76.73
Couple Maximum [N·m]	97.6	180.4	89.4	24.2	20.1	21.3

Tableau Ann.II.3 : Paramètres des actionneurs.

I.5 Références

[1] T. Madani, 'Différentes Approches de Commande Décentralisée à Structure Variable Appliquées en Robotique', thèse de magistère de E.N.P, Alger, 2002.

[2] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the Puma 560 arm," in Proc. International Conference of Robotics and Automation, vol. 1, Washington, USA, pp. 510–18, 1986.

[3] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices", Journal of Applied Mechanics, vol. 77, pp. 215–221, June 1955.

[4] D. Orin, R. McGhee, M. Vukobratovic, and G. Hartoch, "Kinematics and kinetic analysis of open-chain linkages utilizing Newton-Euler methods," Mathematical Biosciences International Journal, vol. 43, pp. 107–130, Feb. 1979.

[5] http://fr.wikipedia.org/wiki/Tenseur#Notion_de_tenseur.

[6] http://fr.wikipedia.org/wiki/Torseur.

TP N°1
Manipulation et Commande du robot à 4ddl
IBM 7576

1. Description du matériel :

Le robot utilisé pour ces travaux pratiques, représente un robot manipulateur à 4ddl, appelé 'IBM 7576' (Figure 1). Il représente le successeur du modèle populaire 7545. Il est rapide, assisté par ordinateur et fabriqué par la société 'Sankyo'. Les caractéristiques du robot sont représentées dans le tableau 1.



Figure 1 : Robot IBM 7576

Marque:	IBM
Modèle:	7576
Type:	Scara
Axes:	4
Précision:	800 mm
Capacité:	10 Kg
Utilisation:	Assemblage

Tableau 1 : Caractéristiques de l'IBM 7576

Ainsi, le système de commande est composé de :

1. la partie de puissance (*host*) et de de la Figure 2, dont un pupitre est détaché ;
2. la commande du robot (*PC AT*),
3. un PC (*roll-up*), permettant d'éditer, de compiler et de simuler les programmes ;
4. un clavier de commande (*Pendant*), permettant d'agir directement sur les axes et sorties du robot.

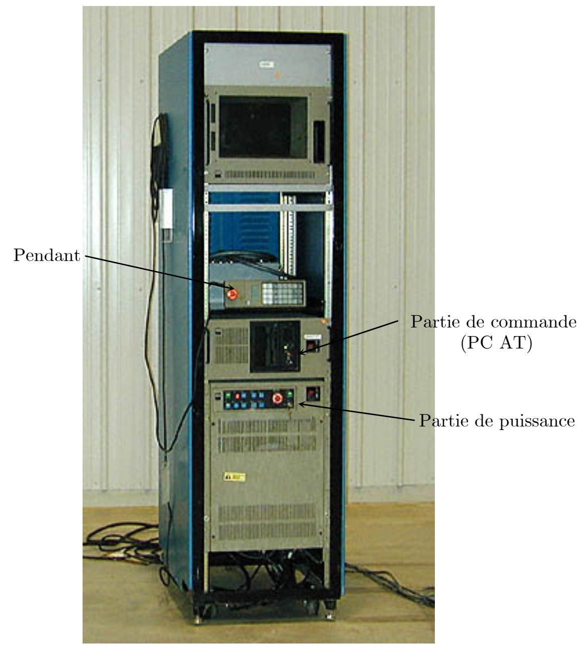


Figure 2 : Armoire de puissance
et de commande du robot IBM 7576

2. Mise en marche :

Avant de manipuler le robot, on doit suivre les étapes suivantes :

- mettre la partie puissance et commande sous tension ;
- taper MENU puis confirmer dans le chemin C:\AML2 ;
- choisir éditer, correspondant au menu 6, puis sur créer (menu 1) pour ouvrir un nouveau fichier (exp : manip1) ;

3. Manipulation avec le Pendant (1 heure):

Pour manipuler le robot en manuel, dans la partie de puissance, il faut se mettre en mode *Teach*, afin de pouvoir découvrir les différentes fonctionnalités offertes dans le **Pendant** (pour plus d'informations sur les touches, veuillez vous référer à la documentation correspondante).

On trouve 48 entrées/sorties présentent dans le système robotisé, dont 24 [1-24] entrées et 24 [25-48] sorties.

- Déterminer le numéro de l'entrée qui représente l'état de la lecture du capteur de chariot (capteur activé indiquant la présence du chariot) ;
- Déterminer la sortie décrivant l'état de la pince (1=ouverte ; 0=fermée) pour chacun des robots étudiés ;
- Déterminer les sorties des trois ventouses du robot 2;
- Déterminer les sorties de chacun des vérins pour les trois robots IBM.

Travail à faire *: déplacer un plot d'un chariot mobile à un plateau fixe.*

Pour importer les coordonnées du point terminal dans votre programme, appuyer sur F7.

4. Initiation à la programmation du robot (1h00 heures):

Les instructions les plus utiles pour le TP 1 sont :

point1 : new pt(x,y,z,α);

/* création d'un point nommé 'point1' de coordonnées (x,y,z) et d'angle de rotation α de la pince*/

programme :subr(a,b,..) ;

/*création d'un sous-programme appelé 'programme'. Au mois un seul sous-programme doit exister dans le programme principal et portant le même nom*/

speed=0.1 ;

/*c'est la vitesse maximale à choisir pour le mouvement du robot*/

writeo(n °sortie, valeur) ;

/*Écrit la valeur (0 ou 1) dans la sortie spécifiée*/

testi(n ° entrée,[valeur, label]) ;

/*retourne la valeur de l'entrée spécifiée et fait un saut à l'étiquette 'label' si l'entrée vaut *valeur*/

pallet(ll,lr,ur,n,N) ;

/* définit une palette de limites 'll,lr,ur,n', dont on accède aux emplacements grâce à ***palletpt(pal,i)****/

Le reste des commandes sont détaillées dans la documentation associée.

Expliquez le code du programme ci-dessous, rédigé en langage AML2 ;

```
Pt1 : new pt(274,286,10,7.5) ;
Pt2 : new pt(0,0,70,0) ;
manip: subr();
speed=0.1;
If testi(1) eq 1 then
begin
pmove(Pt1);
pmove(Pt1+Pt2);
writeo(25,1);
pmove(pt1+<0,0,-30,10>);
writeo(25,0);
end;
end;
```

A partir de ce dernier programme, **écrire un programme qui permet de faire un mouvement vertical et linéaire tout en gérant l'ouverture et la fermeture de la pince ou bien la ventouse.** Utiliser un éditeur DOS ou bien celui du AML2 pour saisir votre code.

Compilez le programme précédent en saisissant les caractères *SY* sur l'invite des commandes. Lorsque le programme est correctement compilé, il faut le télécharger dans le contrôleur (menu 3) en indiquant un numéro d'application (1-7).

Sur le module de puissance, passer en mode 'Auto' puis sélectionner le numéro de votre application, en la téléchargeant grâce au bouton 'Load'. Lorsque cette application est téléchargée (arrêt du clignotement), vous pouvez démarrer l'exécution en appuyant sur 'Start'. Garder le doigt sur le bouton d'arrêt d'urgence afin d'éviter des éventuelles collisions. Le bouton 'End', arrête automatiquement l'exécution du programme après la fin du cycle en cours.

5. Programmation avancée sur le robot (1h00 heures):

On souhaite que le robot réalise des opérations de déchargement de plots métalliques. Pour cela il faut :

1. Détectez la présence d'un chariot mobile après l'avoir bloquer et indexer en activant les soties correspondantes ;
2. Déchargez les plots du chariot et placer les sur un emplacement de la palette fixe. Utilisez les deux fonctions ***pallet(ll,lr,ur,n,N) ;*** *et* ***palletpt(pal,i)*** (voir la documentation du robot IBM), pour définir et accéder au emplacement d'une palette ;
3. En attendant le passage du chariot, le robot doit réaliser la trajectoire de la Figure 3.

Utiliser la documentation fournie pour découvrir les fonctions liées à la palette.

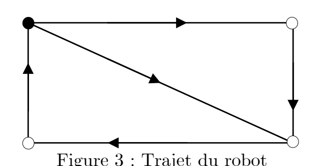


Figure 3 : Trajet du robot

6. Coopération de robots (1h00 heures):

En s'inspirant des exercices précédents, programmez un second robot IBM pour effectuer l'opération inverse du premier, à savoir le chargement des plots de la palette fixe vers le chariot mobile. Exécutez les deux programmes en parallèle ?. en dehors de la tâche demandée au deux robots, chacun d'eux doit réaliser la trajectoire de la Figure 3.

TP N ° 2 Simulation et Commande d'un Robot à 6ddl ABB IRB140

Introduction :

Le robot IRB 140 de la Figure 4 est un robot manipulateur articulé, rigide à six degrés de libertés. Afin d'assurer une grande robustesse et garantir une précision constante même lors de contraintes extrêmement élevées, l'axe 2 utilise la technologie éprouvée ABB de double palier. Ses 6 axes de travail confèrent au robot un rayon d'action exceptionnel, permettant le développement de nouvelles opportunités dans le monde de l'automatisation flexible.

L'IRB 140 peut être fixé au sol, au mur, dans tous les angles et même suspendu. Extrêmement compact avec une hauteur de 800 mm et un poids de 98 Kg, il s'intègre parfaitement aux postes de travail de petites dimensions. Avec sa haute vitesse et sa précision de positionnement de 0,03mm, il est prédestiné au chargement et déchargement de machines, à l'usinage et au montage de petites pièces dans l'industrie horlogère, de la microtechnique et de la machine outil, au chargement d'inserts et au déchargement des presses à injecter, au traitement de surface, ainsi qu'aux opérations de picking et d'emballage. Sa capacité de charge de 5 Kg permet à un préhenseur d'insérer et de décharger des pièces au cours du même cycle, ce qui accroît la vitesse de production.



Figure4: Robot ABB IRB-140

- **L'armoire de commande S4Cplus du robot IRB 140**

La toute génération de commande S4Cplus de la Figure 5, livrée avec le robot IRB 140, encore plus puissante que ces prédécesseurs, constitue une excellente plate-forme pour la création de nouvelles fonctionnalités grâce à l'utilisation du langage de programmation d'applications très puissant et convivial d'ABB, le Rapid. Cette commande, intégrée dans une armoire compacte de dimension 950 x 800 x 620 mm et de poids 250 Kg, conjugue compacité, conception modulaire et haute performance.

La fiabilité, qui a fait l'objet d'une attention toute particulière dans la conception de la commande S4Cplus, permet un fonctionnement précis et sans faille même dans un milieu ambiant avec une humidité relative de 95% et des températures ambiantes de 52 ˚C (valeurs normales dans certaines applications, tel que fonderie). Des variateurs entièrement scellés, une réduction de 35% des points de soudure, une unité de climatisation séparée et un blindage particulièrement soigné garantissant une immunité élevée contre les perturbations électromagnétiques, assurent à la commande et au robot un temps moyen de bon fonctionnement (MTBF) remarquable de 50 000 heures.

Les deux interfaces réseaux Ethernet permettent l'intégration de PC pour la supervision et les réglages des procédés. Autre avantage, le langage et la modularité du système autorisent l'ajout de nouvelles fonctions et l'adaptation aux besoins de chacun.

Le PC industriel équipant la S4Cplus, comprenant un microprocesseur de type multiprocesseur Pentium, un Bus PCI RAM dynamique et un disque Flash, permet la commande simultanée jusqu'à 12 axes (6 axes rotatifs robot et 6 axes externes), avec un fonctionnement à choix dans les modes dynamique, auto optimisé, interpolage d'axes, chaîne de coordination à 7 cadres, concept angulaire ou handling automatique.



Figure 5 : Armoire de commande S4Cplus

- **Logiciel de simulation et d'application convivial et simple d'utilisation**

L'outil logiciel robot de base appelé ***Robot Studio Lite 4.0*** de la figure 6, permet à l'utilisateur d'intégrer des fonctions spécifiques aux applications. Le langage ouvert et la modularité du système simplifie le rajout de nouvelles fonctions ainsi que l'adaptation des programmes existants. Avec les fonctionnalités conviviales du dernier logiciel, le dialogue homme-machine dans un langage pseudo codé avec des messages très clairs et l'écriture de programmes robots, même complexe, est très simple. A l'aide d'un PC raccordé sur le réseau et de ***ProgramMaker*** de la figure 4, l'outil de programmation ABB, ces programmes peuvent être écrit, adaptés et testé hors-ligne avant d'être chargés sur la commande S4Cplus.

De nouveaux outils de simulation développés sur Windows 98/NT pour la commande S4Cplus simplifient la programmation d'applications et la visualisation du process en 3D. D'autre part ces logiciels de simulation permettent de tester des mouvement complexes dans des milieux encombrés et d'obtenir ainsi des données précises sur les temps de cycle. Un setup convivial et rapide facilite l'utilisation des logiciels, et permet au robot IRB 140 de s'adapter rapidement aux opérations de travail. Si, lors de nouvelles contraintes de production, l'utilisation de programmes complémentaires est nécessaire, ceux-ci peuvent être chargés en cours d'utilisation sur la commande du robot. Un logiciel anti-collision avec accouplement de sécurité électronique hypersensible évite toute détérioration lors de fausses manipulations.

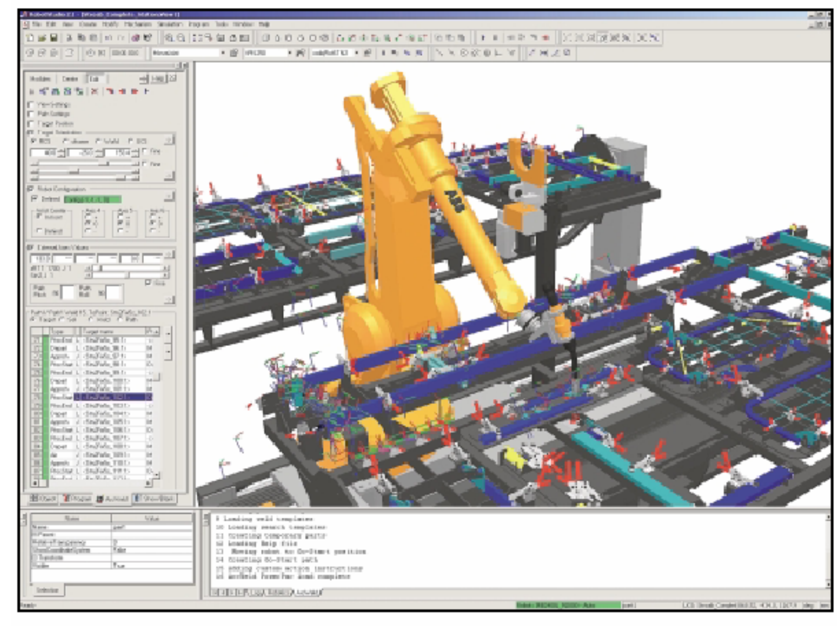


Figure 6 : le Logiciel Robot Studio

Interface de communication étendue

Seules des commandes pouvant dialoguer avec leur environnement avec de multiples technologies, et équipées des systèmes de communication tel que le protocole TCP/IP sur une interface Ethernet avec un réseau PC, ont dans les systèmes d'architecture actuels une chance.

La commande S4Cplus équipée de série avec jusqu'à 1024 entrées/sorties numériques, ne gère non seulement le robot avec plus de précision et une vitesse augmentée, mais offre surtout un large éventail de communication avec d'autres commandes, automates programmables, PC et bus de communications. Elle peut échanger des signaux digitaux en 24 Volts continu ou 120 Volts alternatifs, comprend 3 liaisons série, 2 interfaces pour réseau Ethernet pour 10/100 Mbits/s avec E/S déportées pour API, bus de terrain CAN/Device Net, Interbus-S et Profibus DB, ainsi que des interfaces process pour support et signaux sur le bras supérieur du robot.

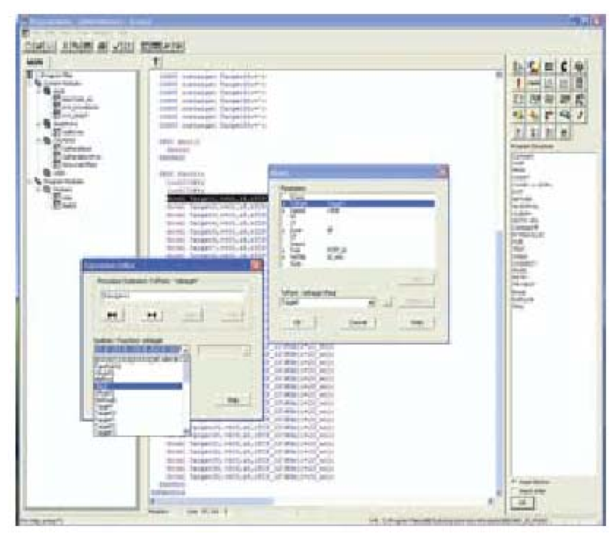


Figure 7 : le logiciel Program Maker

Caractéristiques générales

IRB 140, le mini-robot de ABB pour des temps de cycle court et une qualité irréprochable des pièces produites :

- 6 Axes rotatifs ;

- Grand rayon d'action de 810 mm ;

- Capacité de charge 5 Kg ;

- Encombrement réduit ;

- Version unique pour montage au sol, au mur ou suspendu au plafond ;

- Hauteur : 800 mm ;

- Masse totale : 98 Kg ;

- Charge admissible : 5 Kg ;

- Version salle blanche : 10 ;

- Degré de protection standard / fonderie : IP 54 / IP 67 ;

- Commande robot : S4Cplus.

- **Applications et utilisations**

IRB 140, un robot compact et puissant idéal pour une large gamme d'applications :

- Chargement et déchargement de pièces et outils sur machines de précision ;

- Chargement d'inserts et déchargement des presses à injecter ;

- Usinage, traitement de surface, découpage et ébavurage ;

- Technologies de soudage ;

- Montage et assemblage de composants en salle blanche ;

- Picking, emballage et conditionnement.

Objectifs du TP:

A travers ce TP, on devra pouvoir manipuler le robot **IRB 140** de la Figure 4 de deux manières différentes : manuelle ou automatique, en utilisant soit le joystick du pupitre mobile d'apprentissage de la Figure 5 ou par la réalisation des différentes tâches au robot dans un premier temps, dans un environnement virtuel (Studio Lite, Figure 6), et dans un second temps sur le robot réel en transférant le fichier de simulation, réadapté après compilation grâce à ***Program Maker*** (voir Figure 7) au système de commande S4C.
La durée du TP est de 4 heures, réparties suivant les tâches demandées comme suit :
1. Manipulation et commande du robot **IRB 140** par le biais du pupitre mobile (1 heure) ;
2. Simulation graphique du robot **IRB 140** sous Studio Lite 4.0 en définissant des trajectoires (1 heure) ;
3. Exécution d'une tâche de préhension (1 heure) ;
4. Coopération des ABB (1 heure).



Figure 8 : Exemple de Manipulation du Robot IRB 140

Exercice1 : Manipulation et commande du robot IRB 140 par le biais du pupitre mobile :

Dans cette partie, vous allez manipuler le robot à partir du pupitre :
- Découvrez les fonctions de manipulation manuelle et automatique du robot ABB ;
- Découvrez les entrées/sorties du système. *Combien d'E/S sont présente dans le système ? et à quoi correspondent-elles ?*
- Faite manipuler le robot avec le joystick avec différentes vitesse. *Comment peut-on récupérer les coordonnées du point terminal par rapport à la base selon les coordonnées cartésienne et articulaire ? ;*
- Réaliser la tâche suivante au robot : faite transporter un plot du chariot mobile au plateau fixe. *Quelle est la trajectoire optimale et sûre pour réaliser le dernier mouvement?*

Exercice 2 : Simulation graphique du robot IRB 140 sous Studio Lite 4.0 :

Cinq étapes à réaliser pour cette partie :
- Reproduction du robot IRB 140 et son environnement en 3D ;
- Choix des points de trajectoire (Targets) ;
- Définition des trajets (Paths) ;
- Définition et exécution du programme ;
- Lecture du code avec Program Maker ;
- Synchronisation des programmes et exécution des simulations.

- **Reproduction du robot IRB 140 et son environnement en 3D :**

Dans cette étape, vous devez reproduire graphiquement le robot et son environnement. Pour cela, dans l'espace de travail en 3D du Robot Studio (Figure 9), insérer tous les objets nécessaires pour la représentation du robot et de son environnement.

Dans la fenêtre **File**, cliquez sur **Import** puis **Library** et choisissez le répertoire **enviro-abb1** ou **enviro-abb2**, où ils contiennent une série de fichiers (.rlb) d'objets tels que :
- le robot virtuel **IRB 140** (exp : IRB140-M2000.rlb) ;
- la palette ;
- la pince ;
- la table ;
- le tapis ;
- le chariot mobile ;
- le porte palette ;
- le plot fixe ;
- le plot mobile.

Pour que la pince soit visible, glisser son icône dans celle du robot (c-à-d Irb 140-M2000) ;

Naviguer dans l'environnement virtuel grâce à votre souris :
- deux clicks simultanés sur le bouton du milieu et sur le bouton de droite de la souris, vous permettent de naviguer panoramiquement ;
- un click sur le bouton du milieu et une translation en avant, vous permettent de zoomer en avant ;
- un click sur le bouton du milieu et une translation en arrière, vous permettent de zoomer en arrière ;

La liste de tous les objets présents dans l'espace 3D se trouve dans la fenêtre 'Object', à gauche de votre écran.

- **Choix des points de trajectoire (Targets)**

Une fois le dessin est réalisé, il faut sélectionner les points relatifs aux déplacements du robot, en créant des 'Targets' (Figure 9). Dans un premier exemple, vous allez choisir un point au centre de l'emplacement du plot sur la palette, et un point localisé au centre d'un plot sur le chariot mobile ainsi :

- Pour choisir avec une précision vos points, faite un click droit avec la souris à l'emplacement de la cible et choisissez 'Examine'. L'espace du point sera zoomé ;
- Dans la fenêtre 'Create' du menu, choisissez 'Target'. Une fenêtre de coordonnées de translation et de rotation s'ouvre. Ces coordonnées correspondent à celles du centre d'outil. Ensuite, une fois que vous cliquez sur le bouton gauche de la souris à l'endroit choisi, les coordonnées du point sélectionné vont être calculées par rapport au centre de l'outil (CDO). Validez sur 'Apply' et réadapter l'image à son format initial grâce à 'Unexamine' ;

• Vérifiez que les axes du repère dont l'origine et le CDO ont la même direction avec leurs équivalents des deux points choisis. Sinon, modifiez le sens de leurs orientations en cliquant à droite avec la souris sur l'icône 'Target', suivi de 'Modify' puis'Position'.

- **Définition des trajets (Paths)**
  Dans cette étape, vous devez définir le chemin des trajectoires (Figure 9). Pour cela :
  - Choisissez dans la fenêtre 'Create', le chemin 'Path' ;
  - Glissez les Targets créés dans l'espace de 'Path' ;
  - Remarquez l'apparition d'un trait entre les différents points dans la fenêtre graphique ;

- **Définition et exécution du programme**
  - Activez le contrôleur entre **Robot Studio**, après un click droit sur l'icône **Irb140-M2000** du robot, puis sur **Start Controler**. Une ligne en vert apparaîtra en bas de la fenêtre d'exécution, montrant l'exécution en mode automatique ;
  - Sur l'interface de **Robot Studio**, dans l'espace **Program**, cliquez à droite sur **Program Modules** et créez un nouveau **module**, ensuite un nouveau **main** ;
  - Dans **Robot Studio**, synchroniser les points et la trajectoire simulés avec le contrôleur virtuel, en cliquant sur 'Synchronise' du 'Path' crée;
  - Activez l'option 'Anti Collision' dans la fenêtre graphique (**robot collision**) ;
  - Exécuter ainsi le programme du robot virtuel ABB ;
  - Que remarquez-vous ?

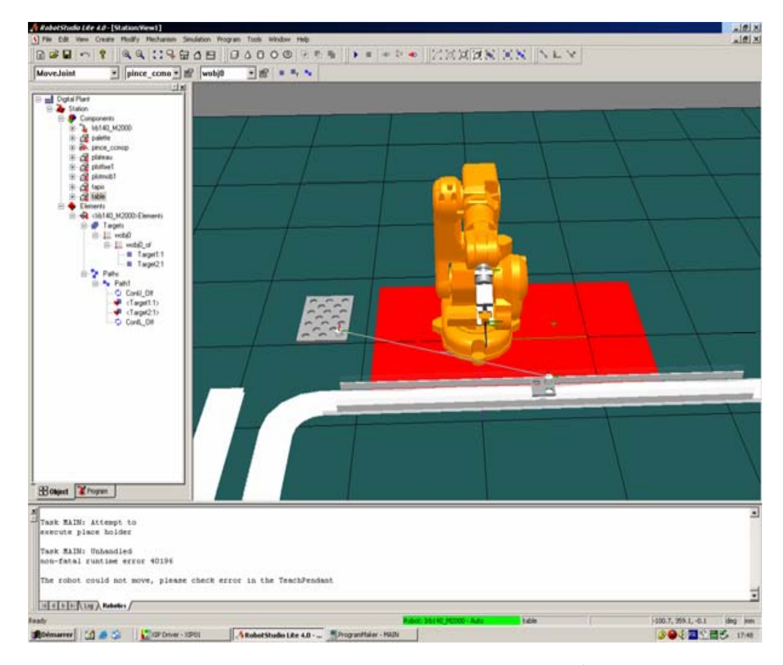


Figure 9 : Planification des trajectoires pour le Robot ABB

- **Lecture du code avec Program Maker**
  - Pour visualiser le code de l'exemple précédent, vous devez lancer **Program Maker 4.0** (figure 10);
  - Dans le Program Maker, vous remarquez l'apparition du programme correspondant au trajet simulé dans **Robot Studio**. Vous pouvez ensuite modifier le code. Les syntaxes des différentes fonctions utilisées sont présentes sous formes d'icônes dans le logiciel et dans le manuel d'utilisation de l'ABB présent à l'AIP ;
  - Expliquez brièvement le code affiché ;
  - Modifiez le programme précédent afin d'éviter les différentes collisions du robot avec son environnement, en rajoutant deux points intermédiaires. Utilisez l'instruction **MoveL** (Move Linear) pour les mouvements verticaux;
  - Compiler votre programme modifié sous **Program Maker** ;
  - Dans **Robot Studio**, à partir de l'anglet **Program**, synchronisez le code modifié à partir de **Program Maker** au contrôleur virtuel, en faisant un click droit sur le Module du programme approprié ;
  - Exécuter de nouveau le programme, que constatez-vous ?

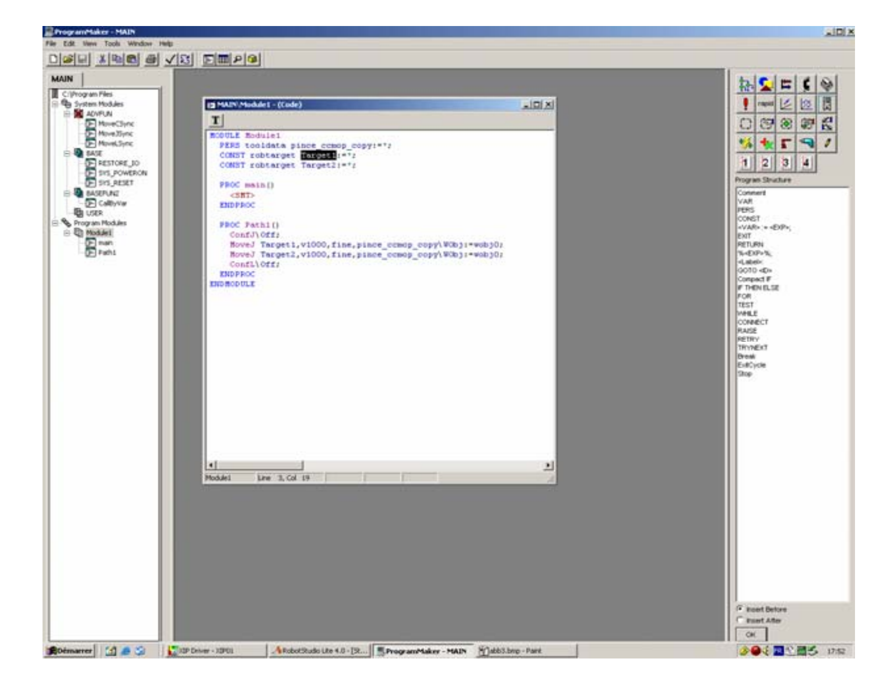


Figure 10 : Programmer sous Program Maker

- **Exécution du programme de simulateur sur le robot réel :**
  - Une fois le simulateur testé, sauver le programme sous (.prj) dans l'invite de commande de **Program Maker**, ensuite envoyez le à travers le réseau à l'ordinateur de l'ABB correspondant en utilisant le programme de **FTP** (Figure 11 ;
  - Ouvrez le programme à partir du pupitre du robot ;
  - Exécuter le programme sur le robot réel en utilisant le pupitre mobile, en mode pas à pas, puis en mode automatique.

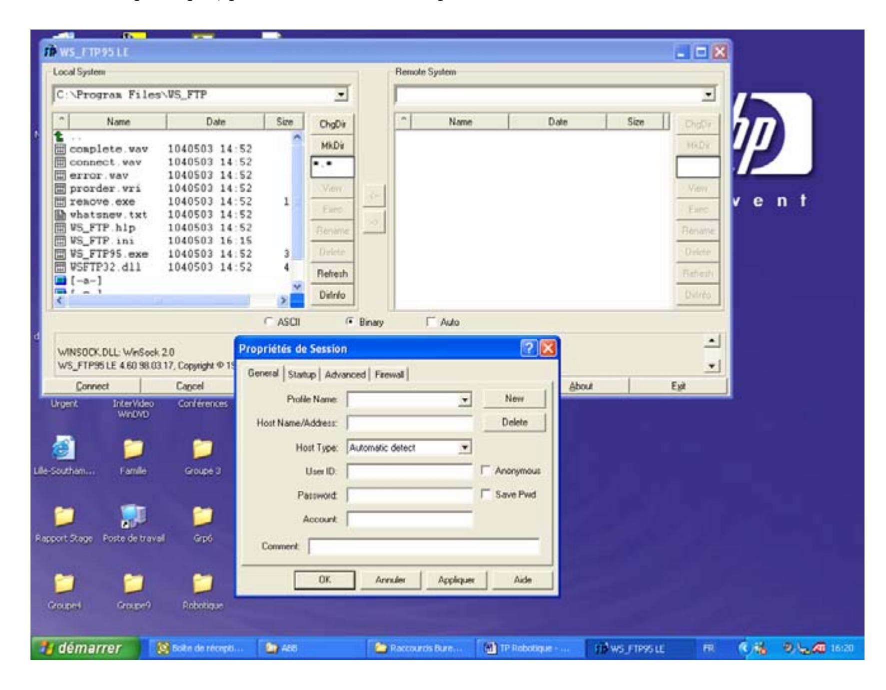


Figure 11 : Fenêtre d'exécution FTP

Exercice 3 : Déchargement de pièces :

En utilisant le même principe de l'exercice 2 (simulation et expérimentation), réalisez une tâche au robot ABB qui permet de :

- Bloquer le passage d'un chariot mobile après avoir actionner un vérin bloqueur (activer la sortie correspondante) ;
- Détecter la présence d'un chariot mobile (lire l'entrée correspondante **WAITDI**) ;
- Fixer le chariot mobile (activer la sortie correspondante) ;
- Décharger trois plots successivement du chariot vers la palette ;
- Libérer le chariot mobile.

Remarque : en attendant l'arrivée du chariot, le robot doit réaliser au dessus de la palette le schéma de l'étoile de la Figure 12.

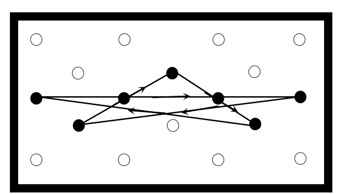


Figure 12 : Etoile en plots

Exercice 4 : Coopération des deux ABB :

Dans cet exercice, vous devez programmer un deuxième robot ABB, qui effectuera la tâche inverse de celui programmé à l'exercice 3, à savoir le déchargement de trois plots consécutifs de la palette vers le chariot mobile. La gestion des entrées-sorties doit être pise en considération, ainsi que la tâche de fond de la Figure 12 lorsque le chariot est absent.

Remarque : au lieu de définir plusieurs points sur la palette, vous devez dans ce cas définir une procédure *Palette*, qui permet de déduire une matrice d'emplacements à partir d'une origine et d'une distance constante entre deux emplacements consécutifs sur les axes X et Y. Pour cela, pensez à utiliser les fonction CPos (permet de lire les données de la position actuelle du CDO) et Offs (permet de déplacer la position du robot).

Exp : pos1 := CPos(\Tool:= toll1\wobj :wobj0) ;

La position courante du CDO soit (x,y,z) est stockée dans la variable pos1

MoveL Offs (p_1,0,0,20), v1000, 250, tool1;

Le robot est amené sur un point qui se trouve à 20mm de la position p_1 dans la direction de Z. Des exemples sont données dans la documentation du robot (3-CPos-1, 3-Offs-1, 3-Offs-2).

TP N ° 3 :
Manipulation et Commande d'un robot à 7ddl
KUKA KR6 ARC



Caractéristiques Techniques du KR 6 ARC

Avec le KR 6 ARC, diverses tâches de précision peuvent être réalisées grâce à son optimisation pour l'Arc, telles que, la soudure, la peinture, l'aspiration, etc.
Ce dernier peut supporter une charge allant de 6 kg jusqu'à 10 kg, dont l'espace de tâche est de l'ordre de 1611 mm.

Il est constitué de 6 axes soit 6ddl avec une option d'un 7ème axe linéaire. Son poids est de 235kg et il peut se fixer sur un sol ou un axe linéaire.

Exercice 1 : Manipulation du Robot KUKA KRC6-ARC

Faite déplacer le robot dans ces sept degrés de liberté, en utilisant le clavier du pendant ainsi que la sourie (pages 15-24 du manuel de formation KUKA), à partir des quatre repères de configuration possibles **(Axe par Axe, WORLD, TOOL, BASE)**. Commentez ?

Exercice 2 : Création d'un centre d'outil avec la méthode d'apprentissage

Le but de cet exercice est de créer un **CDO (Centre D'Outil)** approprié à l'application et l'outil préhenseur. Pour cela, nous allons utiliser la pince à **3 Mors**, équipée d'un objet pointu (exp : crayon), qui nous aidera à mieux localiser le CDO.

1. Vérifier dans la liste des outils, les numéros disponibles à partir de **Configurer, Définition d'outil, Type d'outil, Vérifier la liste des outils**.
2. Allez ensuite sur **Service, Mesurer, Outil**, pour donner un nom à votre CDO (exemple : **Festo-pince** avec une étiquette de **4**). Enfin, pointez le point désiré dans l'espace à partir de quatre orientations différentes, en choisissant la méthode **XYZ-4-Points**.
3. Activer ainsi l'outil initialisé à partir de **Configurer, Outil / Base actuels**.
4. Vérifier et comparer les coordonnées de votre nouveau CDO avec ceux de l'outil appelé **Changeur** déjà crée et qui représente la partie basic de l'outil.

Exercice 3 : Définition d'une base utilisateur

Dans cet exercice, vous allez définir une base utilisateur par rapport à la tâche du robot.

1. la 1ère base utilisateur sera une plaque horizontale fixée sur la table de travail. Allez dans le menu **Service, Mesurer, Base**, et choisissez la méthode 3-points. Cette dernière permet de définir un repère de base à partir de trois points distincts.
2. Dans la 1ère étape, vous devez donner un nom (exemple : **plaque**) et un numéro à votre base. La 2ème étape consiste à choisir l'outil préhenseur avec lequel vous devez définir votre base (prenez celui défini dans l'exercice 2). La 3ème étape permet de choisir l'origine de votre base après placement de l'extrémité de votre pince (CDO) sur un coin de la plaque. La 4ème étape permet de choisir un point sur l'axe X positif (placement>5cm). Enfin la 5ème étape permet de choisir un point dans le plan XY avec Y>0. Sauvegardez ainsi votre base.

3. Refaire les deux points précédents en choisissant la base sur la plaque inclinée (nom : **plaque inclinée**).
4. Vérifiez les coordonnées articulaires et cartésiennes du robot par rapport à sa nouvelle base, à partir de **Visualiser, Position, Cartésien** ou **Articulaire**. Que constatez-vous ?
5. Donnez le sens des axes X, Y, Z dans les nouvelles bases ?
6. Affichez les valeurs incrémentales des 7 positions ?

Exercice 4 : Identification des Entrées/Sorties

A partir des menus C**onfigurer, E/S** et **Technologie, GRIPPER Tech**, listez les entrées et sorties déjà déclarées ?

Exercice 5 : Programmation du robot

Dans cet exercice, vous allez programmer le robot KUKA en planifiant une trajectoire, afin de réaliser des opérations de chargement, déchargement et montage d'une pièce. Cette pièce est composée de trois parties comme le montre la Figure 13.

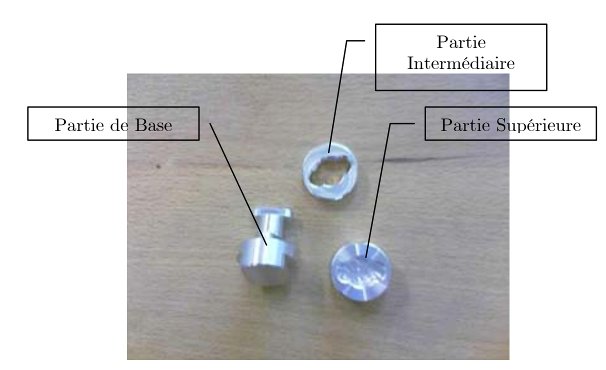


Figure 13 : Structure de la pièce à

Les deux parties intermédiaire et supérieure sont positionnées initialement sur une palette correspondante à leur forme géométrique (Figure 14).



Figure 14 : Palette

Pour écrire un programme à l'aide du pendant, il faut revenir sur le navigateur, ouvrir le répertoire Polytech et ouvrer un nouveau module de programme (page 104).

Les étapes du programme sont décrites comme suit :

1. positionnement initial du robot (**HOME**) ;
2. déplacement du robot jusqu'à la position '2' perpendiculaire à la partie de base localisé dans un chariot mobile (Figure 15). Pour cela, utilisez l'instruction **PTP (point to point)**, pages 73-75 du manuel. Cette instruction permet d'optimiser le déplacement entre deux point tout en optimisant le nombre de degrés de liberté. Pour appeler cette instruction, allez dans le menu **Instructions, Mouvement, PTP**;

3. attente la présence d'un chariot mobile à son emplacement, pour cela il faut bloquer le passage du chariot par le vérin et le fixer à partir de l'indexeur. Pour cela, il faut activer les sorties correspondantes grâce à l'instruction **OUT (Instructions, Logique, OUT, OUT**, page 111**)** et attente d'une entrée grâce à l'instruction **WAITFOR (Instructions, Logique, OUT, WAITFOR**, page 110**) ;**

4. descente verticale vers la partie de base (position '3') après avoir ouvert la pince. La descente étant linéaire, vous allez donc utiliser l'instruction **LIN (Instructions, Mouvement, LIN)**, définie dans les pages 81-84, ainsi que pour activer l'ouverture ou la fermeture de la pince, vous devez utiliser l'instruction **SET (Technologie, GRIPPER Tech, Préhenseur**, pages 117-118);

5. le robot doit prendre ainsi la partie de base et revenir à la position '2', en faisant un mouvement linéaire. **Remarque : quand vous faite de l'apprentissage des points en manuel, veuillez affiner la vitesse du robot par rapport à la tâche à réaliser , afin d'éviter d'éventuels dégâts matériels ;**

6. Ensuite il se dirige à un point perpendiculaire à l'emplacement de dépôt de la partie de base sur la palette fixe (position '5'). Pour cela, le robot doit effectuer un mouvement circulaire de la position '3' en passant par une position intermédiaire '4'. L'instruction appropriée est **CIRC (Instructions, Mouvement, CIRC**, pages 87-91**)** ;

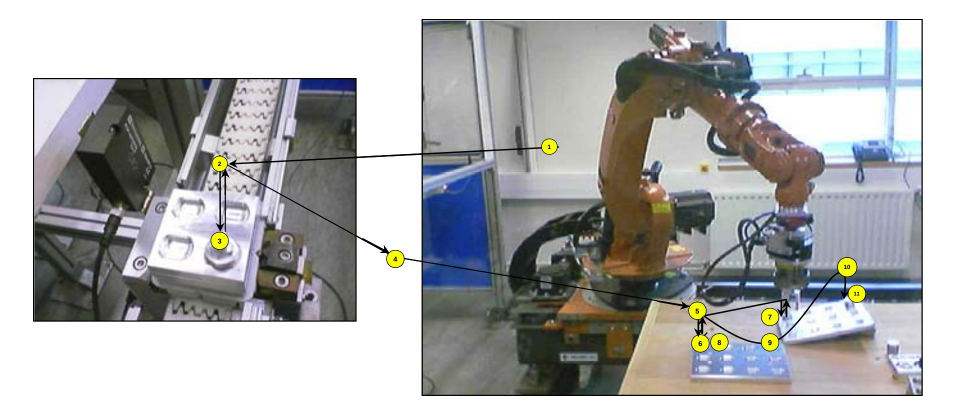


Figure 15: Schéma du montage

7. le robot doit descendre linéairement à la position '6' pour déposer la 1ère partie ;

8. il doit remonter vers la position '5' ;

9. se diriger vers l'emplacement de la 2ème partie de la pièce sur l'autre palette fixe ;

10. prendre la partie intermédiaire et la transporter jusqu'à la position '6' ;

11. positionner la 2ème partie au dessus de la 1ère en affinant la vitesse de descente en mode manuel ;

12. faite un demi tour avec la pince afin de fixer la pièce intermédiaire ;

13. allez à la position '10' au dessus de la partie supérieure (palette 2), en établissant un mouvement circulaire passant par le point '9' ;

14. descendre à la position '11' pour prendre la 3ème partie ;

15. revenir sur le même chemin circulaire et déposer la 3ème partie au dessus des deux précédentes ;

16. prendre la pièce ainsi monter et la déposer sur la table ;

17. exécutez le programme en mode pas à pas puis en mode continu (page 105).